



Modelação, Simulação e Controlo de um Veleiro com Vela Asa

ANA CLÁUDIA BASTOS MIRANDA

novembro de 2021

Instituto Superior de Engenharia do Porto
Departamento de Engenharia Electrotécnica
Rua Dr. António Bernardino de Almeida, 431, 4249-015 Porto

Modelação, Simulação e Controlo de um Veleiro com Vela Asa

Mestrado em Engenharia Eletrotécnica e de Computadores

Ana Cláudia Bastos Miranda

Orientação:

Prof. Manuel Fernando dos Santos Silva
Prof. Maria Benedita Campos Neves Malheiro

Ano Letivo: 2020-2021

Resumo

O objetivo desta dissertação é conceber uma vela asa rígida simétrica, de rotação livre com cauda, integrável num veleiro autónomo com um sistema de controlo baseado em Robot Operating System (ROS). O estudo inicial desta dissertação abrangeu: (i) veleiros autónomos com vela rígida e tradicional e as competições de veleiros autónomos existentes, (ii) os diferentes tipos de materiais e componentes eletrónicos empregues nos veleiros autónomos, (iii) seleção do perfil alar e as forças aplicadas na vela asa e (iv) o estudo das ferramentas de simulação existentes. Em seguida, desenhou-se uma vela asa com cauda (ambas com perfil alar EPPLER 479), um sensor anemómetro (que calcula o vento aparente), um sistema de controlo da cauda (que recorre ao ângulo de ataque atual e desejado como entradas) e um sistema de controlo de direção (que recorre à posição atual do veleiro e à posição desejada como entradas). Posteriormente, procedeu-se à simulação da plataforma desenhada no *software* Autodesk Fusion 360 importada no ambiente de simulação Gazebo. Subsequentemente, foram adicionados diversos *plugins* ao ambiente de simulação, nomeadamente *plugins* reutilizados, alterados e desenvolvidos de raiz. O sistema de controlo foi desenvolvido no ambiente ROS recorrendo a um controlador *Proportional - Differential* (PD) para controlo da cauda e um controlador *Proportional* (P) para controlo da direção do veleiro. Por fim, foram realizados testes de controlo da vela asa, para validar o controlo via cauda e a propulsão do veleiro, assim como testes de controlo de direção para validar o mecanismo de direção do veleiro. Além disso, foi criado o diagrama polar para o modelo desenvolvido que descreve as características dinâmicas do veleiro.

Palavras Chave

Veleiro, vela asa rígida, vela asa rígida de rotação livre.

Abstract

The goal of this dissertation is to design a symmetrical, free-rotating rigid wing sail with tail, integrable in an autonomous sailboat with a control system based on Robot Operating System (ROS). The initial study of this dissertation covered: (i) autonomous sailing with rigid and traditional sails and the existing autonomous sailing competitions, (ii) the different types of materials and electronic components employed in autonomous sailing ships, (iii) selection of the wing profile and the forces applied to the wing sail, and (iv) the study of existing simulation tools. Then, was designed a wingsail with tail (both with EPPLER 479 wing profile), an anemometer sensor (which calculates the apparent wind), a tail control system (which uses the current and desired angle of attack as inputs) and a steering control system (which uses the current sail position and the desired position as inputs). Later on, we proceeded to simulate the platform designed in Autodesk Fusion 360 imported into the Gazebo simulation environment. Subsequently, several plugins were added to the simulation environment, namely plugins reused, changed and developed from scratch. The control system was developed in the ROS environment using a Proportional - Differential (PD) controller for tail control and a Proportional (P) controller for sailboat direction control. Finally, wing sail control tests were performed to validate the control via the tail and the propulsion of the sailboat, as well as steering control tests to validate the steering mechanism of the sailboat. In addition, the polar diagram for the developed model was created, which describes the dynamic characteristics of the sailboat.

Keywords

Sailboat, rigid-wing sail, free-rotating rigid wind sail.

Conteúdo

Conteúdo	i
Lista de Figuras	vii
Lista de Tabelas	xi
Lista de Equações	xiii
Lista de Listagens	xvii
Lista de Acrónimos	xxi
Agradecimentos	xxi
1 Introdução	1
1.1 Problema	1
1.2 Motivação	2
1.3 Objetivos	2
1.4 Requisitos	3
1.5 Planeamento do Projeto	3
1.6 Estrutura da Dissertação	3
2 Veleiros Autónomos	5
2.1 Navegação à Vela	5
2.2 Navegação Autónoma	7
2.2.1 Veleiros Autónomos com Vela Rígida	8
2.2.1.1 Atlantis	8
2.2.1.2 AROO	9
2.2.1.3 ARC	10
2.2.1.4 Beagle-B	12
2.2.1.5 MOOP	13

2.2.1.6	A-Tirma G2	15
2.2.1.7	ASpire	17
2.2.1.8	Maribot Vane	18
2.2.1.9	SB Met	19
2.2.1.10	Análise dos Veleiros de Vela Rígida	20
2.2.2	Veleiros Autónomos com Vela Tradicional	21
2.2.2.1	FASt	22
2.2.2.2	Pinta	23
2.2.2.3	Avalon	24
2.2.2.4	Análise dos Veleiros de Vela Tradicional	26
2.2.3	Discussão dos Diferentes Tipos de Velas	26
2.3	Competições de Veleiros Autónomos	29
2.3.1	The Microtransat Challenge	29
2.3.2	World Robotic Sailing Championship	30
2.3.3	International Robotic Sailing Regatta	30
2.3.4	Discussão das Diferentes Competições	31
2.4	Conclusão do Capítulo	31
3	Caracterização do Veleiro	33
3.1	Plataforma	33
3.1.1	Materiais da Estrutura	33
3.1.2	Casco	34
3.1.3	Leme	35
3.1.4	Patilhão	36
3.1.5	Vela	36
3.1.6	Mastro	37
3.1.7	Sensores	38
3.1.7.1	Sensor de Vento	38
3.1.7.2	IMU	39
3.1.7.3	GNSS	41
3.1.7.4	Encoder Rotativo	41
3.1.8	Atuadores	41
3.1.9	Unidade de Controlo	44
3.1.9.1	Microcontrolador	46
3.1.9.2	Computador de Placa Única	46
3.1.10	Controlo Remoto e Telemetria	48
3.1.11	Alimentação do Sistema	48
3.2	Análise e Comparação	49
3.2.1	Materiais da Estrutura	49
3.2.2	Casco	49
3.2.3	Leme	49
3.2.4	Vela	49

3.2.5	Mastro	50
3.2.6	Sensor de Vento	50
3.2.7	IMU	50
3.2.8	GNSS	50
3.2.9	Encoder Rotativo	51
3.2.10	Atuadores	51
3.2.11	Microcontrolador	51
3.2.12	Computador de Placa Única	51
3.2.13	Controlo Remoto e Telemetria	52
3.2.14	Alimentação	52
3.3	Conclusão do Capítulo	54
4	Proposta de Modelação e Controlo do Veleiro	55
4.1	Proposta	55
4.2	Vela Asa	56
4.2.1	Seleção do Perfil Alar	58
4.2.2	Forças Aplicadas	59
4.2.3	Ajuste da Vela Asa	61
4.2.3.1	Asa com Canard	62
4.2.3.2	Asa com Cauda	62
4.3	Casco	64
4.4	Sistema de Controlo	66
4.4.1	Controlo da Vela Asa	66
4.4.2	Controlo da Direção do Veleiro	66
4.5	Conclusão do Capítulo	67
5	Ferramentas de Simulação	69
5.1	Ambientes de Simulação	69
5.1.1	UWSim	69
5.1.2	CoppeliaSim	70
5.1.3	Kelpie	71
5.2	Simuladores	71
5.2.1	Morse	72
5.2.2	Gazebo	73
5.2.3	USVSim	73
5.3	Análise das Ferramentas de Simulação	75
5.4	Conclusão do Capítulo	77
6	Ambiente de Desenvolvimento	79
6.1	Introdução	79
6.2	ROS	79
6.2.1	Comunicação	80

6.2.2	Conceitos e Terminologias	81
6.3	Gazebo	83
6.4	Integração Gazebo-ROS	83
6.5	Plugins	84
6.5.1	LiftDragPlugin	85
6.5.2	FreeFloating	89
6.5.3	hector_gazebo_plugins	89
6.5.3.1	GNSS	89
6.5.3.2	IMU	90
6.5.4	usv_flap	90
6.5.5	anemometer	90
6.5.6	usv_control	90
6.6	Conclusão do Capítulo	91
7	Desenvolvimento da Simulação	93
7.1	Simulação da Plataforma	93
7.2	Modelo 3D do Veleiro	93
7.3	Definição do Ambiente de Simulação	97
7.3.1	Linguagens	97
7.3.2	Ficheiros	99
7.4	Controlo	101
7.4.1	Controlo da Vela Asa	101
7.4.2	Controlo da Direção do Veleiro	101
7.5	Conclusão do Capítulo	103
8	Testes, Resultados e Validações	105
8.1	Alinhamento da Vela Asa ao Vento	105
8.2	Controlo da Vela Asa	107
8.2.1	Vela Asa Ajustada através da Cauda	107
8.2.2	Funcionamento do Anemómetro	109
8.3	Controlo de Direção	109
8.3.1	Varrimento de Percurso	110
8.3.2	Varrimento de Área	110
8.3.3	Contorno de Obstáculos	111
8.4	Diagrama Polar	112
8.5	Conclusão do Capítulo	114
9	Conclusões	117
9.1	Resultados	117
9.2	Dificuldades	117
9.3	Trabalho Futuro	118
9.4	Conclusão do Capítulo	118

Bibliografia	121
A Nomenclatura e Termos Náuticos	131
B Ambiente ROS e Gazebo	135
B.1 ROS	135
B.2 Gazebo	136
C Instalação de Plugin	139

Lista de Figuras

2.1	Diferentes tipos de maréações.	6
2.2	Opções para a seleção do sistema de propulsão.	8
2.3	Atlantis com vela de asa rígida (©2001 Gabriel Elkaim).	9
2.4	AROO durante testes num lago em Abril de 2005 (©2008 Mark Neal) [1].	10
2.5	ARC durante a competição The Microtransat Challenge em 2006 (©2008 Mark Neal) [1].	11
2.6	Beagle-B na competição World Robotic Sailing Championship (WRSC) na costa de Matosinhos, Portugal, em Julho de 2009 [2].	12
2.7	Beagle-B durante a competição Microtransat Challenge em 2007 (©2008 Mark Neal) [1].	13
2.8	MOOP1 durante o trabalho inicial de teste e desenvolvimento num lago perto de Aberystwyth [2].	14
2.9	Protótipo A-Tirma G2 durante testes no mar (©2015 Jorge Cabrera Gámez) [3].	15
2.10	Veleiros A-Tirma G1 e G2 [4].	16
2.11	ASPIre em testes de navegação no Sudeste de Mariehamn, em Junho de 2018 (©2018 Anna Friebe).	17
2.12	Maribot Vane a navegar contra o vento em Baggensfjärden, Estocolmo [5].	18
2.13	Veleiro SailBuoy Met (SB Met) a navegar no oceano [6].	20
2.14	Trajeto percorrido pelo SB Met da Terra Nova até à Noruega [7].	21
2.15	FEUP Autonomous Sailboat (FASt) durante a competição WRSC de 2008 em Breitenbrunn, Áustria [2].	23
2.16	Veleiro Pinta [8].	23
2.17	Pinta em construção em janeiro de 2008 [1].	24
2.18	Avalon preparado para navegar no lago de Zurich [9].	25
2.19	Veleiro <i>Inflated Wing Sails</i> (IWS) a navegar [10].	29
4.1	Veleiro proposto.	56

4.2	Arquitetura do sistema.	57
4.3	Forças aerodinâmicas e ângulo de ataque α no perfil alar [11].	58
4.4	Secção do perfil alar EPPLER 479 [12].	59
4.5	Principais características do perfil alar Eppler 479 [12].	59
4.6	Forças na vela rígida ao navegar contra o vento (esquerda) e a favor do vento (direita) [4].	60
4.7	Forças e binários envolvidos no controlo da vela: cauda não acionada (parte superior), cauda acionada (centro) e vela asa e cauda em equilíbrio (parte inferior) [13].	63
4.8	Casco Proton [14].	65
4.9	Sistema de controlo da cauda.	67
4.10	Sistema de controlo do leme.	67
5.1	As principais partes que constituem o UWSim [15].	70
5.2	Arquitetura do sistema Kelpie [16].	72
5.3	Arquitetura do Sistema USVSim [17].	74
5.4	Barco a leme, aerobarco, barco diferencial e veleiro do simulador USV-Sim [17].	75
5.5	Subdivisão do casco em 6 partes [17].	75
6.1	Funcionamento do mecanismo de tópicos ROS [18].	81
6.2	Arquitetura da integração Gazebo-ROS [19].	84
6.3	Diagrama de blocos LiftDragPlugin [20].	86
6.4	Relação entre o ângulo de ataque (α) e o coeficiente de sustentação (C_L) [21].	87
7.1	Modelo do veleiro no formato <i>Unified Robot Description Format</i> (URDF).	95
7.2	Modelo da vela criado no <i>software</i> Fusion 360.	96
7.3	Modelo da cauda criado no <i>software</i> Fusion 360.	96
7.4	Sensor anemómetro.	97
7.5	Sensor anemómetro implementado no veleiro.	97
7.6	Modelo do veleiro no Gazebo.	99
7.7	Sistema de controlo da vela asa.	101
7.8	Fluxograma do controlo da vela asa.	102
7.9	Sistema de controlo de direcção	103
8.1	Alinhamento da vela asa ao vento com direcção de 0°	106
8.2	Resposta do alinhamento da vela asa com ângulo de ataque de 0°	106
8.3	Vela asa controlada pela cauda com direcção do vento de 0°	107
8.4	Resposta temporal do ângulo de ataque da vela asa com vento de 0° ($K_p = 0,5$, $K_d = 0,07$, $AoA = 8,5^\circ$).	108
8.5	Resposta temporal do ângulo de ataque da vela asa com vento de 0° ($K_p = 0,5$, $K_d = 0,02$, $AoA = 8,5^\circ$).	108

8.6	Verificação do funcionamento do anemómetro com direção do vento de 0° e $AoA = 8,5^\circ$	109
8.7	Veleiro a percorrer um determinado percurso.	110
8.8	Veleiro a percorrer um determinado perímetro.	111
8.9	Veleiro a varrer uma determinada área.	111
8.10	Varrimento de uma área.	112
8.11	Veleiro a contornar os obstáculos com e sem perturbações.	113
8.12	Contorno dos obstáculos sem perturbações (lado esquerdo) e com perturbações do mar (lado direito).	114
8.13	Diagrama polar.	115
A.1	Partes principais de um veleiro [22].	131

Lista de Tabelas

1.1	Cronograma do projeto.	4
2.1	Características mecânicas dos veleiros com vela rígida estudados.	21
2.2	Características da fonte de energia dos veleiros com vela rígida estudados.	22
2.3	Características mecânicas dos veleiros com vela tradicional estudados.	26
2.4	Características da fonte de energia dos veleiros com vela tradicional estudados.	26
3.1	Vantagens e desvantagens dos mono-cascos e multi-cascos.	35
3.2	Vantagens e desvantagens das velas rígidas simples e duplas.	37
3.3	Características dos sensores de vento estudados.	40
3.4	Características dos sensores <i>Inertial Measuring Unit</i> (IMU) estudados.	42
3.5	Características dos sensores <i>Global Navigation Satellite System</i> (GNSS) estudados.	43
3.6	Características dos <i>encoders</i> rotativos absolutos estudados.	44
3.7	Características dos servomotores estudados.	45
3.8	Características dos Raspberry Pi estudados.	47
3.9	Potência consumida do sistema.	52
4.1	Dimensões da vela asa e cauda.	64
5.1	Características das diversas ferramentas de simulação.	76
6.1	Dimensionamento de parâmetros.	89
A.1	Descrição das partes principais de um veleiro.	132
A.2	Áreas de embarcação.	132
A.3	Termos náuticos.	133

Lista de Equações

4.1	Força de sustentação	57
4.2	Força de arrasto	57
4.3	Número de Reynolds	58
4.4	Segunda lei de Newton	59
4.5	Força de sustentação com a segunda lei de Newton (1)	60
4.6	Força de atrito	60
4.7	Força de sustentação com a segunda lei de Newton (2)	61
4.8	Ângulo entre o vetor de sustentação e o rumo do veículo	61
4.9	Momento da asa	63
4.10	Razão do coeficiente de sustentação	64
6.1	Razão de comprimento adimensional	88
6.2	Razão da área adimensional	88
6.3	Coefficiente do momento	89

Lista de Listagens

6.1	Exemplo de <i>plugin</i> SDF.	88
7.1	Exemplo de código do modelo SDF	98
7.2	Exemplo de código das juntas SDF	98
7.3	Exemplo de código para o ficheiro do mundo da simulação.	100
7.4	Exemplo de código para carregar o mundo.	100
B.1	Ficheiro <code>.bashrc</code>	136

Lista de Acrónimos

- ADC** *Analog to Digital Converter*. 46
- AHRS** *Attitude and Heading Reference System*. 90
- API** *Application Programming Interface*. 70, 73
- ARC** *Autonomous Robotic sailing Craft*. 10, 11, 13, 14, 21, 22, 34–36
- AROO** *Robot for Ocean Observation*. 9–11, 13, 21, 22, 34–36
- ASPIre** *Autonomous Sailing Platform*. 17
- BSD** *Berkeley Software Distribution*. 72
- CAD** *Computer-Aided Design*. 94
- CAN** *Controller Area Network*. 46
- CPU** *Central Processing Unit*. 51
- DART** *Dynamic Animation and Robotics Toolkit*. 73, 83
- FASt** *FEUP Autonomous Sailboat*. vii, 21–23, 26
- FEUP** *Faculdade de Engenharia da Universidade do Porto*. 22
- GNSS** *Global Navigation Satellite System*. xi, 38, 41, 43, 50, 54, 56, 66, 89, 90
- GPS** *Global Positioning System*. 8, 39, 41, 43, 72
- GUI** *Graphical User Interface*. 70, 83, 85
- I/O** *Input/Output*. 46, 51
- I2C** *Inter-Integrated Circuit*. 46

- IMU** *Inertial Measuring Unit*. xi, 38, 39, 41, 42, 50, 54, 66, 89, 90
- INNOC** The Austrian Association for Innovative Computer Science. 30
- IRSC** International Robotic Sailing Conference. 14, 30
- IRSR** International Robotic Sailing Regatta. 29–31
- ISAF** International Sailing Federation. 66
- ISEP** Instituto Superior de Engenharia do Porto. xxi
- IWS** *Inflated Wing Sails*. vii, 28, 29
- LiPo** *Lithium Polymer*. 52, 54
- MOOP** Miniature Ocean Observation Platform. 13, 14, 21, 22, 36, 37
- MORSE** Modular Open Robots Simulation Engine. 69, 72
- MTL** *Material Template Library*. 93, 94
- NWU** *North-West-Up*. 90
- ODE** Open Dynamics Engine. 73, 75, 83
- OGRE** Object-Oriented Graphics Rendering Engine. 73, 83
- OSG** OpenSceneGraph. 71
- P** *Proportional*. iii, 68, 91, 101–103, 109
- PD** *Proportional - Differential*. iii, 68, 91, 101, 103, 107, 109, 113
- PWM** *Pulse Width Modulation*. 66, 67
- RAM** *Random-Access Memory*. 51
- RC** *Remote Control*. 16, 31
- ROS** Robot Operating System. iii, v, viii, 3, 4, 46, 51, 54, 55, 69–72, 74, 75, 77, 79–86, 89–91, 93, 98, 100, 101, 117, 135, 136, 139
- SB** SailBuoy. 20
- SB Met** SailBuoy Met. vii, 19–22, 30, 35
- SB Wave** SailBuoy Wave. 20

- SBC** *Single-Board Computer*. 46, 51, 54, 56
- SDF** *Simulation Description Format*. 71, 85, 93, 94, 96–100
- SPI** *Serial Peripheral Interface*. 46
- TTL** *Transistor-Transistor Level*. 48
- UART** *Universal Asynchronous Receiver-Transmitter*. 48, 50
- UAV** *Unmanned Aerial Vehicles*. 39
- URDF** *Unified Robot Description Format*. viii, 85, 94, 95, 97–100
- USART** *Universal Synchronous and Asynchronous Receiver-Transmitter*. 46
- USB** *Universal Serial Bus*. 46
- USV** *Unmanned Surface Vehicle*. 72, 73
- UUV** *Unmanned Underwater Vehicles*. 70
- WGS84** *World Geodetic System 84*. 89, 90
- WRSC** *World Robotic Sailing Championship*. vii, 12, 14, 22–25, 29, 30
- Xacro** *XML Macros*. 94, 97, 99, 100
- XML** *Extensible Markup Language*. 71, 82, 90, 93, 94, 98, 99, 118

Agradecimentos

A realização desta dissertação reúne os contributos de várias pessoas, o apoio e carinho destas foram cruciais no desenvolvimento deste projeto. Quero desde já expressar a minha gratidão pois, sem estes incentivos, esta caminhada tornar-se-ia mais difícil.

Aos meus orientadores Manuel Silva e Benedita Malheiro pela oportunidade de desenvolver esta dissertação, por todo o apoio, aprendizagem e disponibilidade.

À minha mãe por estar sempre presente quando as coisas não corriam bem, quando o cansaço era muito e sempre acreditou que conseguiria terminar esta etapa.

Por último, quero agradecer aos meus amigos que acompanharam o meu percurso e aos professores que tive o prazer de conhecer enquanto estudante do Instituto Superior de Engenharia do Porto (ISEP).

Capítulo 1

Introdução

Neste capítulo é apresentado o problema que se procura resolver, a motivação do desenvolvimento do projeto, os objetivos a alcançar e a estrutura da dissertação.

1.1 Problema

Nas últimas décadas, tem havido um aumento considerável do interesse nos oceanos. Os oceanos estão pouco explorados, reservando ainda grandes mistérios e oportunidades, nomeadamente em termos de descobertas e avanços significativos do conhecimento. A exploração do oceano abrange várias áreas, desde a mineralogia e geoquímica à tectónica, geofísica e biologia.

Cada vez mais existe a necessidade de construir dispositivos capazes de explorar autonomamente os oceanos por longos períodos de tempo, com baixos custos de operação e manutenção e que sejam aptos a monitorizar grandes áreas de uma forma contínua através da recolha e análise de dados.

Para suprir esta necessidade, surgiram os veleiros autónomos, dispositivos propulsionados pelo vento que são capazes de desempenhar as tarefas desejadas de forma autónoma. Ao longo dos últimos anos tem-se assistido a um interesse crescente por veleiros robóticos, nomeadamente para o estudo e preservação dos oceanos. A utilização destes veículos tem vindo a permitir alcançar zonas inacessíveis, desenvolver sistemas sofisticados de aquisição e transmissão de dados oceânicos, libertar os operadores humanos de tarefas fastidiosas e os responsáveis de campanhas dispendiosas.

O desenvolvimento de um veleiro autónomo é uma tarefa relativamente complexa, desde a construção mecânica da embarcação até à implementação dos seus sistemas eletrónicos. A navegação à vela depende de condições ambientais im-

previsíveis e de variáveis interrelacionadas, como a trajetória, vento (direção e velocidade) e estado do mar. O desempenho de um veleiro depende da maneira como o leme e a vela são controlados. Na vela convencional, o marinheiro controla o leme de acordo com a trajetória desejada e a vela de forma a maximizar a velocidade de deslocamento da embarcação. O problema do controlo do leme foi parcialmente resolvido há décadas através do piloto automático atualmente usado em embarcações comerciais e recreativas. O controlo da vela consiste no posicionamento da vela de forma a definir com o vento o ângulo ótimo que maximiza a velocidade da embarcação na trajetória pretendida. Neste contexto, os veleiros autónomos são embarcações robóticas que utilizam o vento para propulsão e são capazes de controlar o leme e as velas sem intervenção humana.

Além disso, nos últimos anos tem-se assistido a um aumento significativo de adeptos e participantes nas competições de veleiros autónomos, conduzindo a um desenvolvimento acentuado das tecnologias relacionadas. As tecnologias desenvolvidas procuram melhorar os sistemas de navegação utilizando ferramentas de análise e testes dos algoritmos de controlo, de forma a garantir o melhor desempenho possível das embarcações.

Neste trabalho pretende-se modelar, simular e caracterizar o desempenho de um veleiro autónomo equipado com vela rígida e cauda.

1.2 Motivação

A recolha de dados em mar aberto é uma parte fundamental do processo de descoberta científica e monitorização dos oceanos. Algumas áreas de operação são difíceis de alcançar e apresentam riscos substanciais para os pesquisadores envolvidos. Para superar esses obstáculos, o uso de veleiros autónomos para explorar grandes áreas oceânicas de forma sustentável tornou-se uma séria opção.

Esta dissertação surge da necessidade de elaborar um protocolo de controlo para um veleiro que seja capaz de executar missões durante longos períodos de tempo. Além disso, o interesse em aprender conceitos de navegação à vela é também uma das grandes motivações do desenvolvimento desta dissertação.

1.3 Objetivos

O principal objetivo desta dissertação é, primeiramente, analisar e estudar os tipos de veleiros existentes e a dinâmica de veleiros com velas rígidas com cauda. Posteriormente, projetar uma possível solução e elaborar o algoritmo de controlo de um veleiro autónomo que seja capaz de executar missões durante longos períodos de tempo.

Esta dissertação apresenta os seguintes objetivos:

- modelação da vela rígida;
- desenvolvimento de um veleiro autónomo;
- integração da vela e da cauda no veleiro autónomo;
- desenvolvimento de um sistema de controlo da cauda;
- desenvolvimento de um sistema de controlo de direção.

1.4 Requisitos

Relativamente aos requisitos deste projeto podem-se considerar os seguintes:

- conhecimento do *middleware* ROS de forma a estabelecer comunicação entre o modelo do veleiro e o seu ambiente;
- necessário um computador que execute o sistema operativo Linux porque a versão ROS Kinetic apenas suporta o Ubuntu;
- desenvolvimento recorrendo ao ROS;
- o modelo do veleiro deve ser robusto pois está sujeito a condições atmosféricas adversas;
- o controlo do veleiro deve ser fácil e simples.

1.5 Planeamento do Projeto

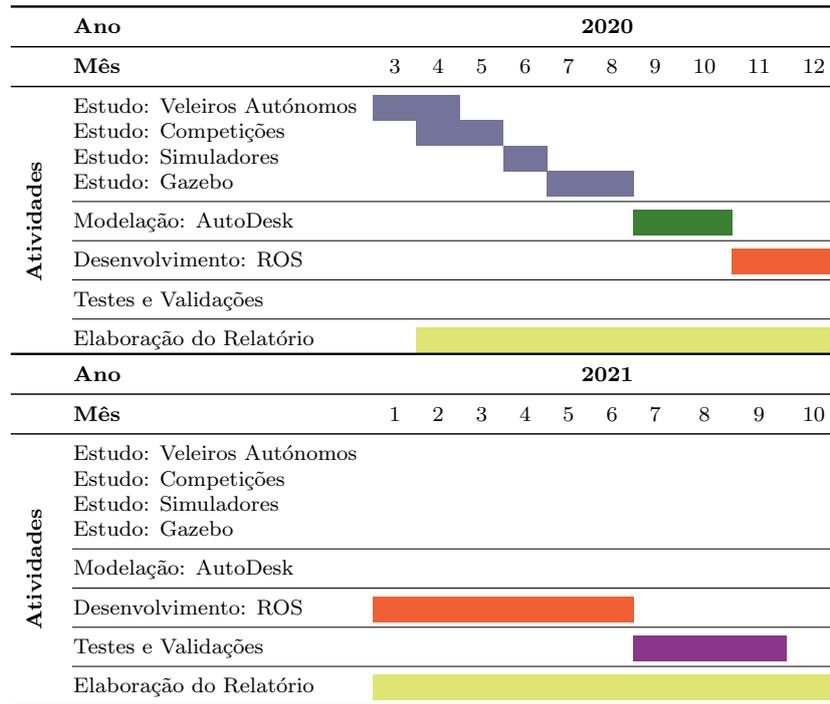
Sendo a modelação, simulação e controlo de um veleiro com vela asa a motivação desta dissertação, e na qualidade do estatuto trabalhador-estudante, a sua prossecução conduziu à calendarização apresentada na Tabela 1.1.

1.6 Estrutura da Dissertação

Esta dissertação é composta pelos seguintes capítulos:

- Capítulo 1 - **Introdução**;
- Capítulo 2 - **Veleiros Autónomos** - analisa os diversos veleiros com vela rígida e vela tradicional e as competições de veleiros autónomos;
- Capítulo 3 - **Caracterização do Veleiro** - detalha os diferentes materiais e componentes elétricos que podem constituir o veleiro;

Tabela 1.1: Cronograma do projeto.



- Capítulo 4 - **Proposta de Modelação e Controlo do Veleiro** - descreve a plataforma proposta, o modelo da vela asa, bem como o sistema de controlo proposto para a cauda e para a direção do veleiro;
- Capítulo 5 - **Ferramentas de Simulação** - analisa as ferramentas de simulação existentes;
- Capítulo 6 - **Ambiente de Desenvolvimento** - aborda o conceito ROS e a comunicação entre o ambiente de simulação. Além disso, apresenta os *plugins* considerados neste dissertação;
- Capítulo 7 - **Desenvolvimento da Simulação** - apresenta o modelo do veleiro 3D, a descrição do ambiente e o sistema de controlo implementado;
- Capítulo 8 - **Testes, Resultados e Validação** - relata os resultados e validação de diversas experiências realizadas para o controlo da vela asa e controlo da direção;
- Capítulo 9 - **Conclusões** - apresenta os resultados da dissertação, a conclusão e o trabalho futuro proposto.

Capítulo 2

Veleiros Autónomos

Neste capítulo são apresentadas as diversas plataformas de veleiros autónomos com vela rígida e tradicional, assim como as competições mais relevantes da atualidade, nomeadamente, The Microtransat Challenge, World Robotic Sailing Championship e International Robotic Sailing Regata.

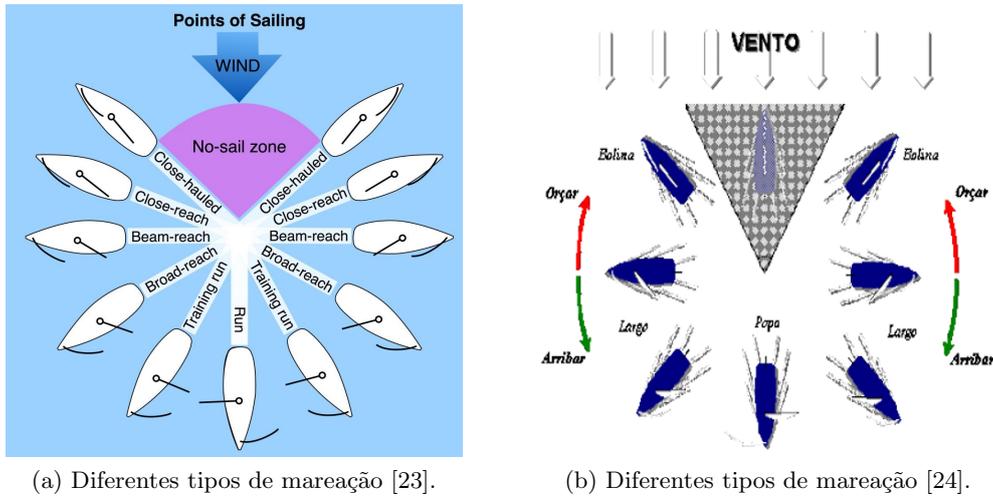
2.1 Navegação à Vela

A navegação à vela pode-se definir como a arte de manobrar as velas de uma embarcação em função do vento (tendo por base a sua direção e intensidade) de modo a que o veleiro execute a sua navegação.

O veleiro quando navega está sujeito a diferentes ângulos de direção do vento. As diferentes posições do veleiro em relação à direção do vento são designadas como pontos de mareação, ilustrados na Figura 2.1. Em cada ponto de partida, é necessário exercer os ajustes necessários nas velas e no leme de forma a executarem a sua função com eficiência.

Os diferentes pontos de navegação, ilustrados na Figura 2.1a, são os seguintes:

- *close-hauled* ou bolina - o veleiro encontra-se o mais próximo possível da linha do vento. As velas estão bem caçadas;
- *close reach* ou bolina folgada - as velas são ligeiramente folgadas;
- *beam reach* ou largo - a linha do vento entra pela alheta do veleiro. As velas são mais folgadas;
- *broad reach* ou largo folgado - o vento entra pela parte traseira, atrás da retranca. As velas são folgadas;



(a) Diferentes tipos de mareação [23].

(b) Diferentes tipos de mareação [24].

Figura 2.1: Diferentes tipos de mareações.

- *training run* ou popa - o vento entra pela ré do veleiro. As velas são folgadas;
- *run* ou popa arrasada - o vento entra pela ré do veleiro. As velas são completamente folgadas e a vela principal é totalmente aberta para o lado oposto ao principal, para que possa levar com o vento. O barco pode continuar a guinar de um lado para o outro.

Relativamente às velas e ao leme, o veleiro atua nestes simplesmente quando as condições, como o vento e a corrente marítima, se alteram, de forma a compensar as rodagens do vento e das correntes. Além disso, o veleiro também atua nas velas e no leme quando pretende deslocar-se para um determinado ponto de referência.

A mareação divide-se em três grandes grupos, ilustrados na Figura 2.1b, que variam conforme a direção da proa em relação à direção do vento, sendo os seguintes: (i) bolina, quando o veleiro navega contra o vento. Existe uma técnica para navegar à bolina empregue por embarcações que consiste em zigzaguear contra o vento, o que permite navegar por zonas onde o vento não é favorável, (ii) largo é quando o vento sopra na parte lateral do veleiro. Nesta situação as velas devem ser um pouco mais caçadas do que na popa. A velocidade alcançada nesta mareação é bastante boa assim como o controlo da embarcação, e (iii) popa quando o vento entra pela ré da embarcação. Neste tipo de navegação a vela grande é toda folgada. Nas embarcações com velas tradicionais, geralmente recorrem ao *spinnaker* (vela balão), em substituição da vela de estai ou genoa, com o objetivo de possibilitar uma maior área exposta ao vento.

2.2 Navegação Autónoma

Os veleiros autónomos executam tarefas complexas de navegação sem qualquer assistência humana. Bowditch [25] define a navegação como “o processo de monitorizar e controlar o movimento de uma embarcação ou veículo de um lugar para outro”.

Segundo Cruz e Alves pode-se definir um veleiro autónomo como “barcos robóticos que usam o vento como fonte de energia para a propulsão e que têm a capacidade de controlar as velas e os lemes sem a intervenção humana” [26].

Os veleiros autónomos possuem espaço livre no seu interior para transportar uma grande variedade de sensores. Assim, segundo [26], os veleiros são capazes de:

- realizar missões de longa duração;
- transmitir dados em tempo real;
- gerar pouco ruído, devido à ausência de motores;
- reduzir os custos operacionais;
- monitorizar grandes áreas de uma forma contínua;
- fornecer dados oceânicos importantes.

Em 2011, Stelzer e Jafarmadar [27] destacaram as principais características de um barco à vela robótico da seguinte forma:

- o barco à vela robótico não é controlado remotamente. Todo o sistema de controlo está a bordo e, por isso, deve executar o planeamento e todas as manobras complexas de navegação de forma totalmente automática e sem intervenção humana;
- o barco à vela robótico é completamente auto-eficiente, isso não é obrigatório no sentido da definição de um veleiro robótico, mas abre uma ampla gama de aplicações.

Na Figura 2.2 pode-se observar as opções para a seleção do sistema de propulsão. Primeiramente, surge a seleção entre uma vela tradicional ou uma vela rígida. Após isso, a escolha da vela concentra-se no tipo de secção, simétrica ou assimétrica. As velas assimétricas apresentam diversas vantagens, tais como o alcance do coeficiente de sustentação máximo e uma razão de C_L/C_D mais alto comparativamente com uma vela de secção simétrica. Ainda assim, decidiu-se

optar pelas velas simétricas dado que um veleiro deve navegar igualmente bem nas amuras de bombordo e estibordo. De seguida, optou-se pela vela simétrica a girar livremente em torno do mastro, ao invés de ser fixa diretamente ao mastro. Por fim, recorreu-se a uma superfície auxiliar para compensar a vela rígida aerodinamicamente de forma a controlar o ângulo de ataque da mesma. Essa superfície denomina-se de cauda. Em suma, as diversas opções direcionaram a uma vela rígida com cauda convencional.

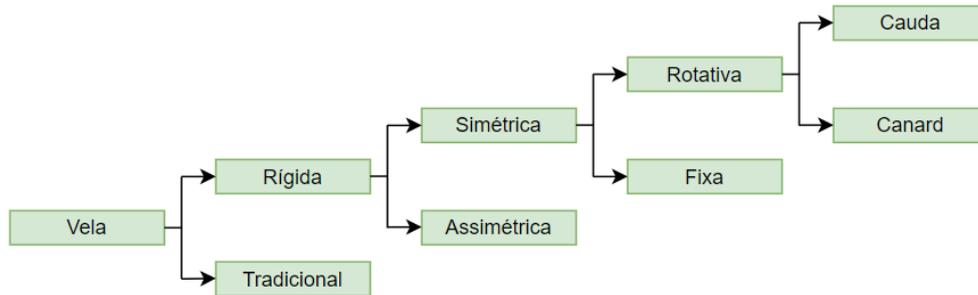


Figura 2.2: Opções para a seleção do sistema de propulsão.

2.2.1 Veleiros Autónomos com Vela Rígida

As velas rígidas permitem que barcos autónomos naveguem com maior autonomia devido ao menor consumo de energia e maior robustez como resultado de um controlo mecanicamente mais simples em comparação com as velas tradicionais. Em contrapartida, as desvantagens das velas rígidas é que não são rizáveis.

Nas próximas sub-subsecções são apresentados vários exemplos de barcos à vela autónomos com vela rígida que foram desenvolvidos ao longo dos anos.

2.2.1.1 Atlantis

O protótipo Atlantis, exibido na Figura 2.3, foi desenvolvido na Universidade de Stanford em 1997, com o conceito de um barco à vela não tripulado, autónomo, conduzido por *Global Positioning System* (GPS).

O Atlantis é um catamarã autónomo de propulsão eólica baseado no Prindle-19 de 7,2 m de comprimento e 3 m de largura, originalmente equipado com uma vela de 17 m² de área [28].

Este projecto tinha como objetivo projectar, desenvolver e testar uma embarcação marítima autónoma. A arquitetura de orientação e controlo do Atlantis permite uma navegação marítima com precisão de pontos de referência, assim como resposta de controlo necessária para uma navegação de precisão na presença de perturbações desconhecidas como vento, correntes e ondas [28].



Figura 2.3: Atlantis com vela de asa rígida (©2001 Gabriel Elkaim).

O sistema de propulsão eólica é uma vela rígida com envergadura de 5,37 m e comprimento de corda de 1,45 m, montada verticalmente em rolamentos auto-compensadores para permitir a rotação livre em torno de um mastro [28].

Vários testes experimentais foram realizados no porto de Redwood City, Califórnia. O catamarã foi sujeito à presença de correntes, vento e ondas. Durante esses testes, o Atlantis foi capaz de demonstrar uma melhoria na precisão de controlo [28].

2.2.1.2 AROO

O Robot for Ocean Observation (AROO), ilustrado na Figura 2.4, foi o primeiro veleiro robótico construído na Aberystwyth University, em 2004 [2]. Este veleiro foi construído com base num casco de plástico de um iate de controlo remoto de 1,52 m de comprimento e, possui uma vela de alumínio com 1,30 m de envergadura e 0,18 m de comprimento de corda.

Este veleiro foi equipado com uma vela rígida devido à sua facilidade de controlo, construção, eficiência e robustez [3]. Além disso, este tipo de vela não entra em colapso com ventos fracos, comparativamente com uma vela tradicional.

O principal objetivo do AROO era provar o conceito de um robô movido à vela com capacidade de navegar durante o maior tempo possível, sob uma ampla gama de condições do vento e do mar.

Surgiu um problema identificado por Sauzé e Neal [1] com este projeto, isto é, a vela poderia girar continuamente e o cabo que liga o sensor de vento ao resto do barco poderia facilmente enroscar-se no mastro. Outros problemas também são mencionados [29]: (*i*) o tempo de resposta do sistema de controlo, uma vez que



Figura 2.4: AROO durante testes num lago em Abril de 2005 (©2008 Mark Neal) [1].

os movimentos das velas eram percebidos como excepcionalmente demorados, às vezes acima de 5 s e (ii) a precisão e manobrabilidade da bússola. Além disso, a vela foi considerada muito grande para o barco causando alguns problemas de estabilidade e dificuldades para o sistema de direção. Sem a possibilidade de rizar a vela, este projeto não seria apropriado para um barco de alto mar onde se espera encontrar ventos acima de 18,01 m/s [35 kn] [1]. Apesar desses problemas identificados, a vela teve um desempenho excepcional com ventos de até 15,43 m/s [30 kn] [1].

O AROO demonstrou a viabilidade básica de operar um robô de navegação autónoma e forneceu a base para trabalhos futuros [2].

2.2.1.3 ARC

O Autonomous Robotic sailing Craft (ARC) foi o segundo veleiro robótico construído pela Aberystwyth University, em 2006. Este veleiro, ilustrado na Figura 2.5, participou na competição The Microtransat Challenge em Toulouse, França, em 2006.

O veleiro ARC tem 1,5 m de comprimento e duas velas rígidas de 1,5 m de envergadura e 0,20 m de comprimento de corda [30]. As velas rígidas do ARC foram construídas com acrílico, tornando-as significativamente mais leves e fáceis de manusear do que as velas do AROO [1].



Figura 2.5: ARC durante a competição The Microtransat Challenge em 2006 (©2008 Mark Neal) [1].

Neste veleiro foram empregues duas velas rígidas controladas de forma independente, com o intuito de combater a instabilidade observada com a vela do AROO, e dois lemes controlados por um único atuador para permitir que o barco navegue em linha reta com pouca ou nenhuma correção de leme (uma vez que os barcos de vela única tendem a desviar-se do rumo). A bússola foi colocada num braço oscilante para a manter nivelada, dado que a bússola no AROO tendia a gerar erros quando o veleiro se inclinava [30].

O ARC é usado principalmente em pesquisas sobre estratégias de sobrevivência autónoma e longevidade em robôs autónomos. É equipado com vários sensores internos que monitorizam o estado do robô, permitindo que ele tome decisões sensatas para manter-se em operação durante um longo período de tempo [30].

Os principais objetivos do desenvolvimento do ARC eram abordar e retificar muitas das deficiências do AROO e introduzir tanta redundância quanto possível.

De acordo com Sauzé e Neal [1], o projeto recorrendo a velas duplas criou uma configuração de navegação muito equilibrada e proporcionou redundância na direção, pois as velas poderiam ser configuradas para fornecer controlo direcional, caso o leme falhasse.

Relativamente ao casco do ARC, este apresentava um comprimento semelhante ao do AROO, mas era mais largo para tornar o barco mais estável e mais fácil de controlar. E, embora Sauzé e Neal [29] estejam plenamente conscientes de que os tamanhos de casco usados no AROO e ARC não são ideais para uso em mar aberto, e que um casco maior seria mais apropriado, eles alegam que, para fins de desenvolvimento e do ponto de vista logístico e financeiro, essas dimensões

foram escolhidas principalmente para permitir que o barco fosse movido por uma única pessoa e se encaixasse dentro de um carro para facilitar o transporte [29].

Segundo Suazé e Neal [1] o desenvolvimento de um veleiro que praticamente não requer o uso do atuador para se manter no curso atual, mantendo assim o consumo de energia num valor absoluto mínimo, é um dos principais requisitos de um robô à vela.

2.2.1.4 Beagle-B

O Beagle-B, ilustrado na Figura 2.6, foi construído para a Aberystwyth University no final de 2006 e início de 2007 pela empresa francesa de robótica Robosoft [31]. A Robosoft foi responsável pela construção do casco, da vela e pela identificação da bússola, GPS, sensor de vento, atuadores e controlador de motor [2].

O veleiro Beagle-B é um mono-casco de 3,5 m de comprimento [27] destinado a velejadores com deficiência, exibindo um projeto particularmente estável e projetado para se auto-corrigir rapidamente em caso de adornamento [1]. Este veleiro é equipado com uma vela de fibra de carbono, além disso possui painéis solares e baterias de chumbo-ácido que, teoricamente, deverá ser suficiente para permanecer no mar continuamente.



Figura 2.6: Beagle-B na competição WRSC na costa de Matosinhos, Portugal, em Julho de 2009 [2].

Relativamente às velas, ainda que a experiência da Aberystwyth University tenha demonstrado que as configurações de vela dupla são preferíveis, o casco do Beagle-B não era adequado para duas velas, e o projeto resultante com meramente uma vela rígida provou ser suficientemente estável. O Beagle-B teve duas versões [32]: (*i*) possuía uma vela rígida de fibra de carbono de 3,0 m de envergadura e 0,85 m de comprimento de corda, no entanto Sauzé e Neal [1] consideraram

esta vela rígida muito grande para velejar sob condições extremas, e elaborou-se uma outra versão (*ii*) que foi equipada com uma vela rígida menor com 2 m de envergadura. A vela rígida do Beagle-B provou ser altamente estável e capaz de navegar com ventos de 0,51 m/s [1 kn].

O principal objetivo do Beagle-B foi ultrapassar muitas das limitações do AROO e do ARC e produzir um barco de grande escala que seria capaz de permanecer no mar por longos períodos de tempo, fornecendo uma plataforma oceanográfica confiável.

O Beagle-B, ilustrado na Figura 2.7, participou na competição The Microtransat Challenge de 2007 e foi capaz de navegar com sucesso por mais de 25 km numa única missão, demonstrando a eficácia do projeto da vela rígida.

Um dos problemas identificados com o Beagle-B era que, pelo menos, duas pessoas seriam precisas para montar e lançar o barco e que em águas movimentadas era sempre necessário um barco de auxílio suficientemente rápido [1].

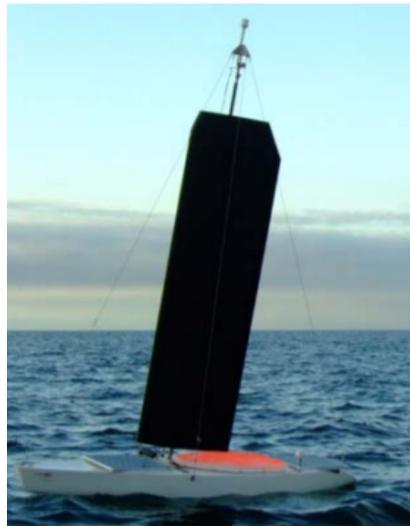


Figura 2.7: Beagle-B durante a competição Microtransat Challenge em 2007 (©2008 Mark Neal) [1].

2.2.1.5 MOOP

O Miniature Ocean Observation Platform (MOOP), representado na Figura 2.8, foi dos mais recentes veleiros desenvolvidos pela Universidade de Aberystwyth, em 2008.

Os veleiros MOOP são pequenos robôs de vela leve, com um comprimento total de 0,72 m e com um peso total de aproximadamente 4 kg.

O objetivo do desenvolvimento destes veleiros era construir um conjunto de robôs pequenos, simples, leves, baratos, fáceis de transportar e fáceis de implementar, mas altamente robustos, que podiam ser produzidos em grandes números, destinados a missões de curto prazo, mas também capazes de atravessar o Atlântico [2]. O baixo custo e o processo de construção simples permitiram produzir um novo barco em menos de três semanas. Isso gerou uma série de variações no projeto original do MOOP, incluindo veleiros com velas duplas, sem lemes, com uma variedade de modelos de sensores de vento e mesmo sem sensores de vento [2]. A lógica era produzir um grande número de plataformas MOOP, que custariam o mesmo que um barco maior, podendo haver uma maior probabilidade de que pelo menos um deles atravessasse o Atlântico. Várias plataformas MOOP participaram nas competições International Robotic Sailing Conference (IRSC) e World Robotic Sailing Championship (WRSC) em 2009 e 2010.



Figura 2.8: MOOP1 durante o trabalho inicial de teste e desenvolvimento num lago perto de Aberystwyth [2].

Anteriormente, o ARC e o Beagle-B tinham a intenção de desempenhar esse papel, no entanto, os MOOP eram vistos como preferíveis porque ofereciam a oportunidade de vários barcos e evitavam as dificuldades operacionais do uso do Beagle-B [2].

Embora a experiência anterior da Universidade de Aberystwyth com o ARC tenha demonstrado que as configurações de vela dupla são preferíveis, o tamanho pequeno do casco dificulta a colocação de duas velas e, portanto, as várias plataformas do MOOP foram equipadas com uma vela rígida, com uma envergadura de 0,525 m e um comprimento de corda de 0,13 m e devem ser pequenos o suficiente para navegarem com ventos fortes. As únicas exceções são o MOOP₃ e o MOOP_n que possuem uma vela dupla para permitir direcionar o barco sem a necessidade de um leme. Essas velas são construídas a partir de um composto de

poliestireno e fibra de vidro resultando numa maneira relativamente barata, fácil e rápida de construir velas [3].

Uma variedade de configurações diferentes foram testadas nas diversas versões do MOOP [2], com a intenção de testar novos projetos de sensores de vento, controlo sem leme e descobrir a direção do vento sem recurso a um sensor de vento. Durante esse processo, foram identificadas várias limitações: (i) dificuldade em navegar contra o vento, (ii) problemas ao navegar em correntes ou marés fortes devido às baixas velocidades do casco e (iii) dificuldade em navegar contra o vento em modelos de vela única.

Em relação aos testes com o MOOP de velas duplas, descobriu-se que era possível dirigir o barco para qualquer ponto da vela, mas que era difícil manter o rumo quando navegava a favor do vento nas versões sem leme. Os projetos de velas duplas mostraram-se mais estáveis em percursos a favor do vento e oferecem o potencial de velejar sem leme ou, pelo menos, que o leme e a vela possam ser usados cooperativamente [3].

2.2.1.6 A-Tirma G2

O veleiro A-Tirma G2, representado na Figura 2.9, tenta superar algumas limitações encontradas no protótipo A-Tirma G1, especialmente adotar um comportamento melhor em condições adversas.

O A-Tirma G2 foi concebido para navegar no oceano num amplo conjunto de condições climáticas. Este projeto foi focado na robustez e na obtenção de algum grau de redundância nos componentes críticos, como velas e leme [4].



Figura 2.9: Protótipo A-Tirma G2 durante testes no mar (©2015 Jorge Cabrera Gámez) [3].

O projeto de um veleiro não tripulado, com o objetivo do A-Tirma G2, implica levar em consideração muitos fatores específicos para torná-lo adaptável e flexível o suficiente para situações distintas sem intervenção humana [4].

O primeiro protótipo, A-Tirma G1, tinha melhorias a serem consideradas, nomeadamente o espaço limitado que possuía. Este veleiro foi baseado num barco *Remote Control* (RC) comercial de baixo custo com comprimento total de 1 m. O A-Tirma G2 é maior em tamanho, com comprimento total de 2 m, permitindo carregar mais equipamento a bordo. Além de que, um comprimento maior, melhora o comportamento de navegação do veleiro, sendo menos influenciado pelos impactos e ondas produzidas [4]. Relativamente ao seu projeto físico, este é desenvolvido a partir do zero e apresenta duas velas de fibra de carbono com envergadura de 1,05 m e corda média de 0,225 m e dois lemes inclinados. A sua curva de estabilidade é totalmente positiva, portanto, o A-Tirma G2 é capaz de se recuperar autonomamente de uma viragem de quilha [4]. Na Figura 2.10 estão ilustrados o A-Tirma G1 e o A-Tirma G2.



Figura 2.10: Veleiros A-Tirma G1 e G2 [4].

No desenvolvimento do veleiro A-Tirma G2 optaram por duas velas rígidas pelos seguintes motivos: (i) permitem propulsão no caso de quebra de uma delas, (ii) duas velas menores, com superfície equivalente a uma maior, produzem menos momento de adorno, (iii) menor consumo de energia com binário reduzido necessário para compensá-las, e (iv) uma plataforma com duas velas rígidas também pode ser interessante como recurso final para conduzir o barco em caso de falha total dos lemes [4]. Em relação ao leme duplo, cada um deles atua numa determinada faixa de ângulos de inclinação de cada lado, sobrepondo a sua atuação a 5° em cada faixa [4], permitindo alcançar um objetivo final de longo prazo, mesmo em condições precárias de navegação. Além disso, o uso de dois lemes, permite que o veleiro continue a navegar em caso de quebra de um deles. Na Figura 2.9 pode-se observar o veleiro A-Tirma durante testes no mar.

Segundo Domínguez-Brito [4] os esforços para melhorar a estabilidade direcional em todas as intensidades de vento e ângulos de adorno permitem otimizar o consumo de energia do veleiro.

Relativamente à navegação a favor do vento, as velas rígidas são mais eficientes do que as velas tradicionais porque, nesse ponto de navegação, as velas de tecido produzem impulso inteiramente por arrasto, o que depende claramente da magnitude do vento aparente. Por outro lado, ao navegar a favor do vento recorrendo a velas rígidas, o impulso é obtido na sustentação e, talvez, também no arrasto [4].

2.2.1.7 ASPire

O Autonomous Sailing Platform (ASPire), retratado na Figura 2.11, foi desenvolvido como o primeiro veículo de superfície de propulsão de vela rígida, na Universidade de Ciências Aplicadas de Åland, entre 2016-2018.

O objetivo de longo prazo da Åland Sailing Robots era desenvolver o primeiro veleiro totalmente autónomo capaz de atravessar com sucesso o Oceano Atlântico. Contudo, o ASPire foi elaborado para pesquisa marinha no mar Báltico com o propósito de recolher dados de sensores marinhos (como por exemplo: medições de condutividade, temperatura e densidade). Esta plataforma é adequada para operações em águas rasas, sendo suficientemente grande para transportar a carga útil necessária para medição de sensores marinhos [33].



Figura 2.11: ASPire em testes de navegação no Sudeste de Mariehamn, em Junho de 2018 (©2018 Anna Friebe).

Relativamente às suas características, o casco usado é um barco de quilha de classe 2,4 mR do projeto Norlin Mark III. O comprimento total é de 4,18 m e a profundidade da quilha é de 1,05 m. A vela principal tem 2,8 m de envergadura com um comprimento de corda de 0,74 m. O perfil da vela é simétrico e consiste

em dois lados superiores do perfil NACA 632-618 [33]. A vela de cauda tem uma envergadura de 0,8 m e um comprimento de corda de 0,3 m. As velas rígidas são fabricadas em fibra de carbono e o mastro é de alumínio, com dois rolamentos de esferas de aço inoxidável [33].

A maioria dos veleiros com velas rotativas têm uma ou várias velas de cauda rígidas controladas diretamente por servo motor. Segundo Anna Friebe [33], a adição de uma vela *canard* apresenta vantagens em comparação com a solução da vela de cauda, tais como a possibilidade de obter uma melhor distribuição de peso diminuindo a necessidade de um contrapeso, o que resulta numa plataforma mais leve. Os veleiros que utilizam velas rotativas com vela de cauda rígida controladas diretamente por servo motor são geralmente trimarãs ou catamarãs [33].

2.2.1.8 Maribot Vane

O projeto Maribot Vane, ilustrado na Figura 2.12, está a ser desenvolvido no Laboratório de Robótica Marítima da Escola de Ciências de Engenharia do KTH Royal Institute of Technology.

O objetivo deste projeto é o desenvolvimento de novas técnicas para criar uma plataforma robusta, capaz de suportar condições complexas durante viagens longas sem assistência humana. O veleiro Maribot Vane tem como objetivo geral monitorizar e recolher dados oceânicos para outros campos de pesquisa (oceanografia, meteorologia ou pesca) no mar Báltico [5].



Figura 2.12: Maribot Vane a navegar contra o vento em Baggensfjärden, Estocolmo [5].

Este veleiro apresenta um mono-casco modificado de 2,4 mR, uma vela de perfil NACA0018 com envergadura de asa de 3,5 m, com uma corda da base de 0,9 m e uma corda de topo de 0,45 m, resultando numa área total de asa de 2,7 m² [3]. A propulsão do Maribot Vane é feita por uma vela rígida de rotação livre e auto-ajustável aumentando a sua robustez.

O Maribot Vane inclui um sistema de auto-direção, sendo uma das suas principais características, que permite que o barco navegue num ângulo aparente constante de vento, usando apenas controlo mecânico pelo vento, com o objetivo de obter um consumo nulo de eletricidade quando usado. Dhomé [5] afirma que este sistema se destina a recolher energia sob certas condições. Além disso, também afirma que o princípio de operação do sistema de direção autónoma é semelhante ao de um mecanismo de direção do cata-vento mas, diferentemente dos lemes convencionais encontrados no mercado, não há cata-vento adicional ao barco para rastrear o vento aparente. A energia é um dos principais parâmetros para navegar durante longas missões. Sendo que o objetivo é não consumir eletricidade na maioria das vezes que o veleiro navega, limitando a necessidade de recarregar as baterias e permitindo navegar em áreas onde é difícil recolher energia do ambiente [5].

Em 2017, foram realizados testes com foco no desempenho do veleiro em diferentes condições, na área de Baggensfjärden, no arquipélago de Estocolmo. Durante esses testes, foi confirmado que o Maribot Vane executa as tarefas como planeado e foi demonstrado experimentalmente a capacidade do sistema de auto-direção [5].

Uma vantagem importante da vela de rotação livre é que ela opera usando um único controlo, o que contribui para um sistema robusto, com poucos modos de falha, em comparação com uma plataforma de veleiro tradicional.

2.2.1.9 SB Met

O veleiro SailBuoy Met (SB Met), ilustrado na Figura 2.13, está a ser desenvolvido e comercializado pela Offshore Sensing AS, na Noruega.

O SB Met é um veículo de superfície não tripulado de longa duração, projetado para suportar uma ampla variedade de aplicações oceânicas. Foi o primeiro veleiro a atravessar o oceano Atlântico, vencendo a competição The Microtransat Challenge, em 26 de agosto de 2018, percorrendo um total de 5354 km [34].

Este veleiro tem 2 m de comprimento, pesa 60 kg e é capaz de transportar cerca de 15 kg de carga útil [3]. A velocidade média é de 0,51 m/s a 1,03 m/s [1 kn a 2 kn] e consegue navegar com velocidade de vento entre 2 m/s a 20 m/s. O SB Met utiliza uma vela rígida montada na proa para navegar na direção a pontos de referência predefinidos, tornando-a uma alternativa atraente para bóias de superfície à deriva.

O SB Met possui uma bateria para alimentar o piloto automático interno com o intuito de navegar durante seis meses sem carregar. Este também foi equipado com painéis solares para alimentar os atuadores e a eletrónica. Os sensores e o sistema de comunicação possuem uma bateria separada, com a capacidade de 240 W/h, 20 A/h e 12 V [3].



Figura 2.13: Veleiro SB Met a navegar no oceano [6].

A Offshore Sensing construiu 14 SailBuoy (SB) cobertos por painéis solares que alimentam a tecnologia a bordo, e uma vela trapezoidal rígida montada perto da proa que impulsiona a embarcação.

Os SB podem ser equipados com sensores para medir ondas, salinidade do oceano e níveis de oxigénio. Eles são vendidos por um custo semelhante ao do aluguer de um navio de pesquisa, por alguns dias. E, segundo Peddie [35] “A grande vantagem é que podemos adquirir uma enorme quantidade de dados por um custo muito baixo”.

Para além do SB Met existe também um outro veleiro Sailbuoy, que é o SailBuoy Wave (SB Wave), que atravessou metade do oceano Atlântico em 2017.

O SB Met localizado na Terra Nova navegou para a Irlanda percorrendo um total de 5100 km. Depois de atravessar o oceano Atlântico, o SB Met também alcançou a Noruega. Nesta missão, o veleiro navegou 118 dias no mar e percorreu um total de 7800 km em todos os tipos de clima, desde a Terra Nova até à Noruega, via Irlanda [3]. Na Figura 2.14 é ilustrado o trajeto total percorrido pelo veleiro SB Met. Com esta missão pode-se comprovar que este veleiro é capaz de navegar durante longos períodos de tempo, sujeito a condições climáticas adversas.

2.2.1.10 Análise dos Veleiros de Vela Rígida

Nesta sub-subsecção são apresentadas as principais características mecânicas, resumidas na Tabela 2.1, e as características da fonte de energia, ilustradas na Tabela 2.2, dos veleiros com vela rígida estudados anteriormente.

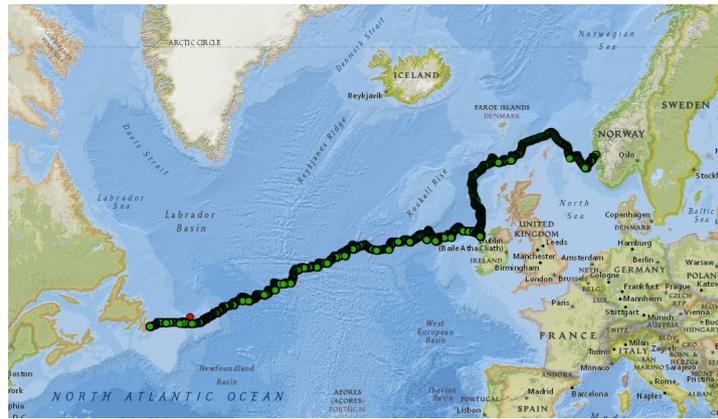


Figura 2.14: Trajeto percorrido pelo SB Met da Terra Nova até à Noruega [7].

Tabela 2.1: Características mecânicas dos veleiros com vela rígida estudados.

Nome	Casco		Vela rígida	
	Tipo	Comprimento (m)	Constituição	Dimensões (m ²)
Atlantis	Bi-casco	7,20	Tela	5,37 x 1,45
AROO	Mono-casco	1,52	Alumínio	1,30 x 0,18
ARC	Mono-casco	1,50	Acrílico	1,50 x 0,20
Beagle-B	Mono-casco	3,50	Fibra de carbono	3,0 x 0,85
MOOP	Mono-casco	0,72	Poliestireno e fibra de vidro	0,525 x 0,13
A-Tirma G2	Mono-casco	2,00	Fibra de carbono	1,05 x 0,225
ASPIre	Mono-casco	4,18	Fibra de carbono	2,80 x 0,74
Maribot Vane	Mono-casco	4,20	Alumínio	3,50 x n.a
SB Met	Mono-casco	2,00	n.a	n.a

(n.a. - informação não disponível ou não encontrada pela autora.)

2.2.2 Veleiros Autónomos com Vela Tradicional

As velas tradicionais permitem que a sua área seja reduzida de maneira relativamente fácil, recorrendo ao rizar convencional ou trocas das velas. Além disso, elas podem ser reparadas e modificadas com facilidade sendo uma das grandes vantagens deste tipo de velas quando comparadas com as velas rígidas. Em contrapartida, as velas tradicionais são propensas a desgastar e rasgar quando configuradas incorretamente, sendo uma das suas desvantagens.

Nas sub-subsecções seguintes são apresentados os veleiros autónomos com vela tradicional que se consideraram mais representativos, sendo estes o FEUP

Tabela 2.2: Características da fonte de energia dos veleiros com vela rígida estudados.

Nome	Fonte de energia		
	Tensão (V)	Corrente (A)	Alimentação
Atlantis	n.a	n.a	n.a
AROO	12,0	4,2	Chumbo ácido
ARC	1,2	2,3	Hidreto metálico de níquel
Beagle-B	12,0	60,0	4 x Chumbo ácido e 2 x painéis fotovoltaicos 45 W _p
MOOP	1,2	2,7	Hidreto metálico de níquel
A-Tirma G2	n.a	n.a	n.a
ASPire	12,0	110,0	Gel e 1 x painel fotovoltaico
Maribot Vane	n.a	n.a	n.a
SB Met	12,0	20,0	240 W h lítio e painéis fotovoltaicos

(n.a. - informação não disponível ou não encontrada pela autora.)

Autonomous Sailboat (FASt), Pinta e o Avalon.

2.2.2.1 FAST

A Faculdade de Engenharia da Universidade do Porto (FEUP) desenvolveu um veleiro robótico conhecido por FAST, representado na Figura 2.15, em 2007.

O seu projeto foi baseado num casco Open 60, mas reduzido a 2,5 m de comprimento, pesa 50 kg e a área total da vela é de 3,7 m².

Este projeto elaborado pela FEUP visava construir um veleiro autónomo de pequena escala com dois objetivos: (i) minimizar a energia necessária para navegar e (ii) navegar o mais rápido possível. Além disso, este veleiro tinha como objetivo participar numa competição internacional de embarcações à vela não tripuladas, intitulada The Microtransat Challenge, assim como realizar missões oceanográficas.

O veleiro FAST é capaz de navegar de forma totalmente autónoma e com um consumo de energia reduzido. Esta é uma plataforma não tripulada capaz de realizar missões planeadas, transportando uma ampla gama de sensores para manobrar o leme e orientar as velas, sendo capaz de permanecer no oceano por longos períodos de tempo [36]. A energia eléctrica é assegurada por um painel solar e armazenada em baterias.

O veleiro FAST participou na competição WRSC nos anos 2008, 2009 e em 2012. Em 2012, o FAST conquistou o primeiro lugar na 5ª competição do WRSC,



Figura 2.15: FAST durante a competição WRSC de 2008 em Breitenbrunn, Áustria [2].

que ocorreu entre os dias 17 e 21 de setembro em Cardiff, Wales [36].

2.2.2.2 Pinta

O veleiro Pinta, ilustrado na Figura 2.16, foi construído pela Universidade de Aberystwyth com a intenção de competir no WRSC e na corrida transatlântica The Microtransat Challenge, em outubro de 2008 [8].



Figura 2.16: Veleiro Pinta [8].

Este veleiro é baseado num veleiro Topper Taz [1], com 2,95 m de comprimento, ilustrado Figura 2.17. E, ao contrário de todos os outros veleiros robóticos

desenvolvidos pela Universidade de Aberystwyth, utiliza uma única vela tradicional, em vez de uma vela rígida, controlada por um motor. Essa decisão foi tomada simplesmente para facilitar a construção e reduzir o custo [2]. No entanto, a vela existente era considerada muito grande, tendo sido cortada para a área total de 2 m^2 [2]. O seu projeto baseia-se essencialmente nas lições aprendidas com os veleiros robóticos desenvolvidos anteriormente.



Figura 2.17: Pinta em construção em janeiro de 2008 [1].

O principal objetivo do Pinta era atravessar o Oceano Atlântico na competição The Microtransat Challenge.

Relativamente aos sensores utilizados, este desfruta do mesmo tipo de computador, sensor de vento, bússola, etc. que o veleiro Beagle-B. E embora este último fosse provavelmente capaz de atravessar o Oceano Atlântico era considerado muito caro para arriscar. Portanto, era necessário um barco mais barato, que replicasse de perto o desempenho do Beagle-B, e que fosse de baixo custo e dispensável.

Em 2010, o Pinta foi o único barco a participar no The Microtransat Challenge [6] tendo percorrido uma distância de 87 km de forma autónoma e 653 km até à última mensagem enviada.

2.2.2.3 Avalon

O Avalon, ilustrado na Figura 2.18, foi desenvolvido por uma equipa de estudantes do Instituto Federal de Tecnologia de Zurique com o objetivo de participar na competição The Microtransat Challenge. Este veleiro participou também no desafio WRSC, em 2009 [27].



Figura 2.18: Avalon preparado para navegar no lago de Zurich [9].

O Avalon possui um mono-casco com 3,95 m de comprimento e um sistema de leme duplo. Este veleiro apresentou um projeto de equipamento equilibrado, com um mastro em forma de L e uma vela de tecido, com uma área total de $8,4\text{ m}^2$ [2].

Segundo Hendrik Erckens [9] foram considerados os seguintes aspetos para o projeto do Avalon:

- altas cargas e forças nas estruturas mecânicas devido a ventos fortes e condições climáticas severas;
- maiores exigências de confiabilidade. Não há chance de reparar nada ao longo da jornada;
- transmissão de força preferencialmente do atuador para a vela, a fim de economizar o máximo de energia possível.

O principal objetivo deste veleiro é atravessar autonomamente o Oceano Atlântico no desafio The Microtransat Challenge, sendo capaz de evitar obstáculos estáticos e dinâmicos como também realizar automaticamente manobras de viragem de bordo e cambar.

Em 2009, durante a competição WRSC, o Avalon sofreu problemas ao fazer curvas grandes, o que causou um esforço excessivo no motor Maxon que controlava a vela, resultando num sobre-aquecimento do motor e danos na caixa de engrenagens [2].

2.2.2.4 Análise dos Veleiros de Vela Tradicional

Nesta sub-subsecção são apresentadas as principais características mecânicas, resumidas na Tabela 2.3, e as características da fonte de energia, ilustradas na Tabela 2.4, dos veleiros com vela tradicional estudados primeiramente.

Tabela 2.3: Características mecânicas dos veleiros com vela tradicional estudados.

Nome	Casco		Vela tradicional
	Tipo	Comprimento (m)	Dimensões (m ²)
FASt	Mono-casco	2,50	3,70
Pinta	Mono-casco	2,95	2,00
Avalon	Mono-casco	3,95	8,40

Tabela 2.4: Características da fonte de energia dos veleiros com vela tradicional estudados.

Nome	Fonte de Energia		
	Tensão (V)	Corrente (A)	Alimentação
FASt	n.a	n.a	2 x Lítio e 1 x painel fotovoltaico 45 W _p
Pinta	12,0	7,0	16 x Chumbo ácido e 6 x painéis fotovoltaicos de 120 W _p
Avalon	25,2	23,8	4 x Lítio-magnésio e 4 x painéis fotovoltaicos 90 W _p

(n.a. - informação não disponível ou não encontrada pela autora.)

2.2.3 Discussão dos Diferentes Tipos de Velas

Nesta subsecção são apresentadas as principais vantagens e desvantagens dos veleiros autónomos com vela rígida e com vela tradicional.

As velas tradicionais são empregues em todo o mundo há milhares de anos em inúmeros barcos. De acordo com Neal *et al.* [37] as velas de tecido têm várias propriedades úteis:

- podem ser convenientemente abaixadas e armazenadas quando estiverem no porto;
- a sua área pode ser reduzida de maneira relativamente fácil, rizando ou trocando de velas;
- podem ser reparadas e modificadas com facilidade;

- a sua forma e curvatura podem ser alteradas, caçando e folgando as cordas de controlo.

No entanto, de acordo com diversos investigadores este tipo de velas tradicionais também apresentam as suas desvantagens [28, 29, 37]:

- as velas tradicionais são propensas a desgastar e rasgar quando manobradas incorretamente;
- devem ser enroladas quando não estiverem em uso, a fim de evitar flagelações destrutivas;
- geralmente são controladas por cordas, que frequentemente quebram ou atolam, e que requerem a atenção regular da tripulação. E, executar essas tarefas de forma autónoma, mesmo que fosse concebível, incorreria em despesas significativas, resultando num uso excessivo de energia, peso e custo financeiro;
- perdem a forma quando não são mantidas com um ângulo de ataque correto, tendo de ser reajustadas – caçadas ou folgadas – para evitar a perda completa de manobrabilidade da embarcação;
- tendem a torcer, levando a diferentes ângulos de ataque em diferentes pontos da vela, o que pode reduzir a eficiência da vela.

As principais vantagens das velas rígidas em comparação com as velas tradicionais são que as primeiras mantêm a forma com ventos fracos, em contrapartida as velas tradicionais perdem a forma. As velas rígidas são mais robustas de controlar porque não dependem de cabos, sofrem menos resistência e têm melhor eficiência.

No entanto, as velas rígidas também apresentam algumas desvantagens, sendo as mencionadas com mais frequência as seguintes [1]: (i) a dificuldade de construir velas rígidas que sejam robustas, leves e baratas, (ii) não podem ser rizadas com o intuito de reduzir o seu tamanho em ventos fortes e (iii) não são particularmente estáveis ao navegar a favor do vento, principalmente em barcos à vela individuais.

Após o estudo de diversos veleiros, verifica-se que há uma variação significativa tanto em termos de formato, tamanho do casco bem como o objetivo do projeto. No entanto, no que diz respeito às velas rígidas existem alguns atributos comuns que podem ser observados, segundo Stelzer e Dalmau [38]:

- as velas devem ser à prova d'água, para que possam sobreviver submersas e ajudar a endireitar o veleiro em caso de emborcamento;

- as velas devem também ser leves para simplificar o armazenamento e transporte;
- se possível, os cabos não devem passar pela vela, se o fizerem, os anéis coletores devem ser usados para permitir a transferência de energia e sinais, ou a rotação da vela deve ser limitada, ou os cabos devem passar através de um tubo fixo;
- o tamanho da vela deve ser pequeno. Ao projetar os barcos para competições, é interessante aumentar o tamanho da vela com o intuito de aumentar a velocidade, mas isso geralmente é contraproducente ao navegar com ventos acima de 15,43 m/s [30 kn], pois isso pode levar a ângulos de inclinação de mais de 45° na maioria das vezes, o que pode diminuir a eficiência do leme se não for projetado adequadamente para essa situação;
- dada a atual falta de mecanismos simples e confiáveis de redução da área vélica (rizar) de velas rígidas, os veleiros que se deparam com ventos fortes devem estar equipados com uma vela pequena;
- uma vela rígida equilibrada é muito mais eficiente em relação ao consumo de energia do que qualquer outra plataforma convencional.

Com vista a melhorar algumas das desvantagens das velas rígidas, como a dificuldade de reduzir a área vélica quando o veleiro está sujeito a condições atmosféricas adversas, surgiu o veleiro *Inflated Wing Sails* (IWS), ilustrado na Figura 2.19, desenvolvido pela Next Technologies.

A criação do protótipo foi em 2005 e no ano 2016 foi feito o pedido de patente internacional e a validação do desempenho do sistema em comparação com uma vela tradicional. Em 2017, foi construído um veleiro equipado com uma IWS de 42 m² e um mastro telescópico de 13 m. O mastro expande-se e retrai-se pneumáticamente. A fase de testes foi ultrapassada com sucesso e toda a navegação mostra que o sistema IWS é simples e eficiente [10].

O IWS é estável em todas as condições e cria muita pouca força na estrutura. Além disso, oferece uma navegação simplificada, suave e equilibrada, que elimina a necessidade de guinchos, adriças, linhas de manivela e acessórios complexos.

A vela deste veleiro é completamente flexível e concebido com: (i) pele dupla, formando um perfil aerodinâmico simétrico, (ii) ventiladores na ponta estabilizando a forma em todas as velocidades e direções do vento e (iii) um mastro retrátil localizado no centro de impulso aerodinâmico do perfil. O IWS possui diversas vantagens tais como [10]:

- a vela voa verticalmente e o perfil NACA é estudado para desenvolver uma forte tração para a frente;



Figura 2.19: Veleiro IWS a navegar [10].

- o uso de um perfil simétrico permite navegar de uma borda para a outra sem ajustar o perfil;
- o perfil simétrico é auto gerido e posiciona-se automaticamente no melhor ângulo de tração;
- o centro de impulso aerodinâmico permanece estável em todas as velocidades do vento.

O sistema IWS é previsto ser empregue em velas de iate de cruzeiro mais simples e refinadas. Este tipo de equipamento retrátil oferece uma abordagem totalmente nova para a navegação. E, para grandes velas de superfície, como os barcos de transporte de passageiros ou mercadorias, possibilita a redução de forças no tecido da vela, no cordame e na estrutura do barco [10].

2.3 Competições de Veleiros Autónomos

Nesta secção são apresentadas as três competições de veleiros autónomos que se consideraram mais significativas da atualidade, sendo estas a The Microtransat Challenge, WRSC e International Robotic Sailing Regatta (IRSR). Estas competições têm o objetivo de promover o desenvolvimento dos veleiros estudados previamente.

2.3.1 The Microtransat Challenge

O The Microtransat Challenge foi concebido por Mark Neal, da Aberystwyth University, e Yves Brière, do Institut Supérieure de l'Aéronautique et de l'Espace em Toulouse, em França, em 2005. A primeira corrida transatlântica deveria ter

ocorrido em Portugal, em 2008, mas ficou adiada para 2010 e o ponto de partida foi alterado para a Irlanda [6].

O The Microtransat Challenge é uma competição para projetar e construir um veleiro autónomo, inicialmente com menos de 4 m e desde 2017 limitado a 2,4 m de comprimento, capaz de atravessar o Oceano Atlântico sem interação humana. Esta competição foi dividida em duas classes [6]: (i) a classe Navegação, que permite apenas energia eólica para propulsão, e o comprimento total da linha de água de carga do casco é restrito a 2,4 m, e (ii) a classe Não Navegação, que permite qualquer fonte de propulsão e até 2,4 m de comprimento total da linha de água de carga do casco.

Em 2018, houve um veleiro que conseguiu ultrapassar o oceano Atlântico, vencendo pela primeira vez a competição The Microtransat Challenge. Este veleiro é o SB Met abordado na sub-subsecção 2.2.1.9.

2.3.2 World Robotic Sailing Championship

A primeira competição WRSC ocorreu em 2008, na Áustria, organizada pela The Austrian Association for Innovative Computer Science (INNOC). Além disso, realizou-se em simultâneo o International Robotic Sailing Conference (IRSC) em Breitenbrunn, no Lago Neusiedl [39]. Desde então, ambos os eventos passaram a ter lugar em conjunto anualmente.

O WRSC [39] é uma competição autónoma de veleiros, derivada do Microtransat Challenge, que tem como objetivo promover o desenvolvimento da robótica marinha através de uma série de regatas, desafios de navegação e autonomia. E o IRSC acompanha o evento, permitindo aos investigadores apresentar e trocar ideias acerca dos diversos problemas científicos envolvidos no projeto e no desenvolvimento dos veleiros autónomos.

O WRSC é uma competição anual, aberta a veleiros totalmente autónomos e não tripulados, com até 4 m de comprimento. Em 2016 [40], esta competição foi dividida em duas classes: (i) a classe Micro Veleiro, para barcos com comprimento total menor que 1,5 m e uma massa até 100 kg, e (ii) a classe Veleiro, para barcos até 4 m e menos de 300 kg de massa. Relativamente às especificações dos veleiros, estes podem usar qualquer tipo de casco (mono ou multi) e qualquer tipo de plataforma, com uma ou mais velas tradicionais ou rígidas. Além disso, os veleiros apenas podem recorrer à energia do vento ou das ondas como fonte de propulsão.

2.3.3 International Robotic Sailing Regatta

O IRSR é uma competição realizada na América do Norte, na qual equipas de estudantes de instituições de ensino superior e escolas secundárias competem. O

primeiro evento foi realizado em 2006, pela Universidade de Queen's, em Kingston, no Canadá [41].

O principal objetivo desta competição é a criação de um veleiro não tripulado, que seja capaz de executar uma variedade de desafios, tais como corrida, navegação, entre outros. Os alunos devem tirar o melhor partido destas competições amigáveis entre escolas e aplicar os seus conhecimentos de engenharia.

O IRSR tem duas classes de embarcações [41]: (i) a classe Sailbot cujos barcos têm até 2 m de comprimento, onde pelo menos metade da equipa deve ser do ensino secundário ou superior, e (ii) a classe Aberta, com barcos até 4 m de comprimento.

Inicialmente, muitas equipas recorrem ao MaxiMOOP como sua primeira plataforma, devido ao seu tamanho reduzido (1,2 m), baixo peso (20 kg), maior capacidade de carga útil do que um barco RC e melhor navegabilidade.

2.3.4 Discussão das Diferentes Competições

Nos últimos anos tem-se assistido a um aumento significativo de adeptos e participantes nas competições de veleiros autónomos, conduzindo a um aumento acentuado na evolução das tecnologias.

As competições são uma componente primordial no desenvolvimento de um veleiro autónomo, uma vez que permitem aos investigadores apresentar e trocar ideias acerca dos diversos problemas científicos envolvidos nos projetos. Assim como, melhorar todos os anos as suas plataformas com a finalidade de alcançar os objetivos das competições.

2.4 Conclusão do Capítulo

Este capítulo apresenta uma pesquisa sobre veleiros autónomos com vela rígida e vela tradicional, com base principalmente na literatura disponível. Para cada veleiro autónomo estudado foram apresentadas as suas principais características, os seus objetivos, bem como as suas vantagens e desvantagens e outros aspetos que se consideraram significativos abordar.

Veleiros autónomos são veículos que apenas utilizam a energia do vento e das ondas como forma de propulsão, com capacidade de controlar as suas velas e lemes sem qualquer intervenção humana.

Com base no estudo anterior, concluiu-se que os veleiros autónomos de vela rígida estão-se a tornar cada vez mais comuns, principalmente devido à sua robustez e eficiência quando comparados com os veleiros autónomos de vela tradicional.

Capítulo 3

Caracterização do Veleiro

Neste capítulo é apresentada uma possível solução de um veleiro autónomo com base nas características dos veleiros estudados no Capítulo 2.

3.1 Plataforma

Nesta secção vai-se abordar os possíveis tipos de materiais, o casco, o leme, o patilhão, a vela, o sistema de controlo e a comunicação. A embarcação tem como principal objetivo ser robusta, uma vez que estará sujeita a condições atmosféricas adversas, viável e simples de controlar.

3.1.1 Materiais da Estrutura

A seleção dos materiais para construir o casco é crucial, dado que tem um impacto direto no peso do veleiro, na resistência e no preço. Além disso, os materiais que compõem a vela e o mastro também são muito importantes. Nesta subsecção, a fibra de vidro, o aço inoxidável, o alumínio, a fibra de carbono e a madeira são apresentados como materiais de construção válidos para o casco, a vela e o mastro.

- fibra de vidro - a fibra de vidro é composta por filamentos de fios de vidro, agregados através de resina, derivados de silício com densidade de 2550 kg/m^3 [42]. A fibra de vidro apresenta baixo teor alcalino, baixo coeficiente de dilatação térmica e baixo custo quando comparado com outras fibras de reforço de alta desempenho [42].
- aço inoxidável - o aço inoxidável é uma liga com altos percentuais de ferro, cromo e níquel com densidade de 7700 kg/m^3 . A principal diferença entre o

aço comum e o aço inoxidável é o facto do último ser resistente à corrosão e a altas temperaturas [43]. Além disso, o aço inoxidável tem um longo ciclo de vida.

- alumínio - o alumínio é um metal leve com densidade de 2700 kg/m^3 . O alumínio puro é um metal macio que, quando ligado a outros metais, como o silício, cobre e magnésio, torna-se um material mais forte [44]. Além disso, este metal é barato, resistente à corrosão e pode ser reciclado várias vezes [45].
- madeira - a madeira é um material produzido a partir do tecido formado pelas plantas lenhosas com funções de sustentação mecânica [46]. A densidade da madeira difere de tipo para tipo, dependendo do ambiente de crescimento da árvore, da espécie da árvore, entre outros fatores. Considerando a madeira cedro, utilizada para móveis, construção naval, entre outras aplicações, apresenta uma densidade de 485 kg/m^3 [47].
- fibra de carbono - a fibra de carbono é uma fibra sintética composta de finos filamentos de 5 a $10 \text{ }\mu\text{m}$ de diâmetro e composta principalmente de carbono com densidade de 1780 kg/m^3 [48]. A fibra de carbono apresenta uma alta resistência mecânica, com alto módulo de elasticidade e baixa densidade, em comparação com outros elementos, como o aço [48].

3.1.2 Casco

O casco de um veleiro é a concha externa do barco que contém todos os componentes internos do veleiro. Existem dois tipos de cascos, o mono-casco e o multi-casco. Os veleiros com cascos do tipo multi-casco podem ser: (i) catamarã, uma embarcação com dois cascos com propulsão à vela ou a motor e (ii) trimarã, uma embarcação com três cascos, dois cascos mais finos situados de um lado e de outro de um casco central.

De forma a determinar qual o melhor tipo de casco a ser empregue no veleiro, elaborou-se o seguinte estudo, ilustrado na Tabela 3.1.

Com base na Tabela 3.1 conseguiu-se elaborar um resumo acerca das principais vantagens e desvantagens dos dois tipos de casco. Os diferentes tipos de casco são úteis consoante as diferentes condições que o veleiro enfrentará. Como dito anteriormente, este veleiro tem como objetivo ser robusto, estável e simples de controlar.

Segundo Sauzé e Neal [29], os tamanhos de casco usados no AROO e ARC não são ideais para uso em mar aberto e um casco maior seria mais apropriado. As dimensões foram escolhidas principalmente para permitir que o barco fosse movido por uma única pessoa e de fácil transporte. O veleiro ARC apresentava um

Tabela 3.1: Vantagens e desvantagens dos mono-cascos e multi-cascos.

	Mono-casco	Multi-casco
Vantagens	Menor custo Aproveita melhor o vento Permite o adornamento	Maior estabilidade Mais espaço para o material Melhor navegabilidade contra o vento Melhor comportamento com marés inconstantes Geralmente, navegam com maior velocidade
Desvantagens	Menos estável Capotamento mais fácil	Custo elevado Exigem mais espaço para virar e atracar

casco semelhante em comprimento comparativamente com o veleiro AROO, mas mais largo, com o objetivo de tornar o barco mais estável e mais fácil de controlar. Tendo por base os veleiros estudados anteriormente, verificou-se que o A-Tirma G2 apresenta um casco maior, comparativamente com o protótipo anterior A-Tirma G1, com 2 m de comprimento, permitindo transportar mais equipamentos a bordo [4]. Um comprimento maior melhora o comportamento de navegação do veleiro, sendo menos influenciado pelos impactos e ondas produzidas [4]. Além disso, o veleiro SB Met, que atravessou o oceano Atlântico, apresenta um casco de 2 m de comprimento e é capaz de transportar cerca de 15 kg.

3.1.3 Leme

O leme é um dispositivo de controlo da direcção de embarcações. O princípio de funcionamento consiste em desviar o fluxo do fluido (água) de modo a que através de um par acção/reacção consiga rodar o veículo para a posição pretendida.

Um veleiro autónomo deve ser capaz de navegar sem intervenção humana. Isto é, todas as tarefas que geralmente são realizadas num veleiro convencional, devem ser executadas de forma autónoma. Para isso, é necessário o manuseio autónomo do leme de forma a que o veleiro efetue o percurso definido autonomamente.

Com base nos veleiros estudados anteriormente, verificou-se que o veleiro ARC empregou dois lemes de forma a permitir navegar em linha reta com pouca ou nenhuma correção do leme. Além disso, possuir dois lemes oferece redundância adicional se um dos motores falhar. Além deste veleiro, o A-Tirma G2 também recorreu ao uso de dois lemes, onde cada um deles atua numa determinada faixa de ângulos. O uso de dois lemes permite que o veleiro continue a navegar em caso de quebra de um deles. Um dos problemas do veleiro A-Tirma G2 era que adornava e, de forma a conceber uma maior estabilidade ao veleiro, decidiram empregar dois

lemes, sendo este um dos principais motivos deste veleiro apresentar dois lemes. Segundo Domínguez-Brito [4] os esforços para melhorar a estabilidade direcional em todas as intensidades de vento e ângulos de adorno permitem otimizar o consumo de energia do veleiro.

O recurso a dois lemes utiliza-se principalmente quando o veleiro apresenta uma boca larga, isto é, quando na ré a largura é elevada e, para permitir melhor equilíbrio do veleiro. Contudo, o recurso de apenas um leme permite um controlo mais fácil e simples, sendo a simplicidade um dos factores primordiais deste projeto.

3.1.4 Patilhão

O patilhão de uma embarcação é o plano vertical submerso no sentido longitudinal que impede uma embarcação de derivar ou abater lateralmente [49].

Relativamente ao patilhão pretende-se um alongamento elevado, pois, quanto mais elevado este coeficiente for, menos força de arrasto longitudinal é gerada, mantendo a força de arrasto lateral suficiente para que o veleiro não derive por ação do vento. Por razões de estabilidade também é benéfico que este coeficiente seja elevado, uma vez que, o patilhão possui um bolbo na sua extremidade, quanto maior este for, mais baixo será o centro de gravidade, melhorando a estabilidade [49].

3.1.5 Vela

A vela é das componentes mais importantes na construção e desenvolvimento de um veleiro autónomo, uma vez que o desempenho do veleiro depende do ajuste da mesma.

Segundo Stelzer e Dalmau [38] as velas devem ser pequenas para navegar com ventos fortes, embora seja intuitivo que ao aumentar o tamanho da vela, aumenta a velocidade do veleiro. No entanto, como dito anteriormente, o objetivo principal deste veleiro é que seja robusto, sendo a velocidade do veleiro um factor secundário.

Tendo por base os veleiros estudados no capítulo anterior, verificou-se que o veleiro ARC apresenta uma vela mais pequena do que a do AROO com a finalidade de combater a instabilidade ao navegar com ventos fortes.

Para além da dimensão da vela, teve-se em consideração quantas velas vão ser empregues no veleiro. Existem diversos veleiros com vela rígida dupla, como é o caso do ARC, e segundo Neal [37] ao recorrer a velas duplas criou-se uma configuração de navegação mais equilibrada proporcionando redundância na direção, pois as velas poderiam ser configuradas para fornecer controlo direcional, caso o leme falhasse. Além disso, o veleiro MOOP apresenta diversos tipos de

configurações, entre os quais recorre às velas duplas e descobriu-se que era possível conduzir o barco em qualquer ponto da vela, mas que era difícil manter o rumo quando navegava a favor do vento nas versões sem leme. Além disso, os projetos das plataformas MOOP de velas duplas mostraram-se mais estáveis em trajetos a favor do vento [3]. O projeto do veleiro que se está a propor partilha do mesmo objetivo do A-Tirma G2, concebido para navegar num amplo conjunto de condições climáticas, e focado na robustez. Por isso, teve-se em consideração o projeto deste veleiro que optou também por velas duplas. Esta opção deveu-se aos seguintes motivos: (i) permitem propulsão no caso de quebra de uma delas, (ii) duas velas menores, com superfície equivalente a uma maior, produzem menos momento de adornamento, (iii) menor consumo de energia, com redução do binário necessário para compensá-las, (iv) uma plataforma com duas velas rígidas pode ser interessante como recurso final em caso de falha total dos lemes. Além disso, ao tomar a decisão do número de velas a utilizar, é fundamental ter em consideração o tamanho do casco, pois um casco pequeno dificulta a colocação de duas velas. Por isso, algumas configurações das plataformas MOOP foram equipadas com uma vela rígida, dado que o tamanho do casco era pequeno.

Com o intuito de determinar quantas velas rígidas vão ser empregues no veleiro fez-se o seguinte estudo, ilustrado na Tabela 3.2.

Tabela 3.2: Vantagens e desvantagens das velas rígidas simples e duplas.

	Velas Rígidas Simples	Velas Rígidas Duplas
Vantagens	Mais fácil de controlar	Maior robustez <i>Station keeping</i> Menor consumo de energia Permitem propulsão no caso de quebra de uma das velas
Desvantagens	Menor robustez	Sincronização das duas velas

Além das vantagens e desvantagens, apresentadas na Tabela 3.2, relativamente ao número de velas a serem empregues no veleiro, é necessário ter em conta a aplicação do mesmo. A vela rígida simples é mais fácil de controlar quando se está perante condições atmosféricas adversas. Em contrapartida, a vela rígida dupla é adequada quando se pretende que o veleiro fique parado no mesmo sítio, designado *station keeping*, num determinado diâmetro.

3.1.6 Mastro

Mastro ou mastaréu é o termo náutico que designa a(s) longa(s) peça(s) vertical(is) que nas embarcações à vela sustenta(m) a retranca, as cruzetas e o velame. Os mastros geralmente têm secção quadrada, circular ou oval e são fabricados em madeira, aço, alumínio ou fibra de carbono [50].

O mastro serve para fixar a vela principal ao veleiro. Na escolha do tipo de mastro pode-se recorrer a duas análises: (i) mastro fixo à base do casco, onde a estrutura da vela gira em torno do mastro e (ii) mastro que gira livremente. Contudo, existem em ambos os casos considerações importantes. No primeiro caso, uma das principais dificuldades é a complexidade no projeto de uma espécie de camisa que permite a vela girar livremente. No segundo caso, o mastro ao girar livremente, deve-se certificar que o mesmo está fixo ao casco e não ocorre nenhuma adversidade, por isso é necessário uma peça que permita assegurar essa fixação, onde o projeto da mesma é uma das maiores complexidades deste tipo de mastro.

3.1.7 Sensores

O desenvolvimento de um veleiro autónomo requer uma seleção de diversos sensores, como *Inertial Measuring Unit* (IMU) e *Global Navigation Satellite System* (GNSS) que fornecem informações acerca da orientação e movimento do veículo. Um requisito comum para a maioria dos veleiros autónomos é um sensor de vento para medir a direção assim como a velocidade do vento. Todas as informações pretendidas, como posição, rumo ou velocidade são obtidas por diversos sensores localizados no veleiro. Além disso, o desenvolvimento de um veleiro autónomo requer unidades lógicas para processar as informações dos sensores e controlar os atuadores, como servo motores. Nesta subsecção são estudados os diversos sensores que podem ser utilizados no veleiro.

3.1.7.1 Sensor de Vento

O sensor do vento fornece a velocidade e a direção do vento aparente. Os projetos de sensores de vento mais comuns são baseados em pás de vento em movimento, embora sejam potencialmente susceptíveis a falhas mecânicas e tenham maior probabilidade de apresentar problemas de confiabilidade. Por esse motivo, muitos veleiros evitam o uso de anemómetro mecânico e, em vez disso, optam por anemómetros ultrassónicos. Embora o sensor de vento seja leve, pequeno e barato, requer um processamento de sinal significativo, principalmente para corrigir desvios dependentes de temperatura. O cata-vento, usado para indicar a direção do vento, é um dos instrumentos meteorológicos mais antigos. Este sensor é montado num eixo ou pináculo elevado, no caso dos veleiros, é montado no cume do mastro, para ter a medida menos perturbada possível. Este sensor é utilizado para encontrar a orientação do vento em relação ao veleiro. De forma a encontrar a orientação absoluta do vento (relativamente ao norte), é necessário adicionar a orientação do veleiro, informação fornecida pelo IMU.

O PCE-FST-200-201 é um anemómetro de 3 copos e é sensível à velocidade do vento de menos de 0,5 m/s a 50 m/s. Este sensor de velocidade do vento está

disponível em diferentes versões com diferentes saídas, ou seja, o PCE-FST-200-201-I com sinal de saída de 0 mA a 20 mA e o PCE-FST-200-201-U com um sinal de saída de 0 V a 12 V [51]. Além disso, apresenta tensão de entrada de 12 V a 36 V. Relativamente às suas dimensões, este sensor apresenta 127,5 mm de altura e 131,5 mm de diâmetro. Da mesma empresa, foi também desenvolvido um outro sensor denominado PCE-FST-200-202. Este sensor mede a direção do vento e os resultados são emitidos na forma analógica, podendo ser montado numa ampla variedade de equipamentos de medição disponíveis, detetando até os menores movimentos do ar e os resultados são transmitidos diretamente [52]. O sensor de direção do vento PCE-FST-200-202 é sensível à velocidade do vento até menos de 0,8 m/s. Este sensor é fabricado em ligas metálicas, resistente à corrosão tornando-o resistente à oxidação ou ferrugem, permitindo que o dispositivo seja utilizado nos ambientes mais difíceis. Este sensor mede a direção do vento com resolução de 22,5° com uma precisão de $\pm 3^\circ$. Ademais, as suas leituras variam de 0° a 360° com uma velocidade mínima de vento de 0,8 m/s e uma velocidade máxima do vento de 70 m/s. A tensão de entrada varia entre 12 V a 36 V com uma resistência elétrica de 250 Ω . Em relação às suas dimensões, possui 157,5 mm de altura e 234,0 mm de diâmetro. O sinal de saída é o mesmo do sensor anterior (0 V a 12 V ou 0 mA a 20 mA) [52].

Os sensores de vento referidos acima detetam a velocidade e a direção do vento, individualmente. No entanto, existem módulos de sensores de vento que são constituídos por um sensor de velocidade do vento e um sensor de direção do vento. Por exemplo, o módulo RK120-01C desenvolvido pela RIKA [53]. Este sensor é montado com peças de precisão em liga de alumínio, apresentando uma elevada resistência. O sensor apresenta uma faixa de medição entre 0 m/s a 45 m/s com uma precisão de $\pm 0,3 + 0,03 Vm/s$, onde V é a velocidade do vento, e um limite mínimo de velocidade do vento de 0,5 m/s. O sensor de direção do vento tem uma faixa de medição de 0 - 360° com uma precisão de $\leq \pm 3^\circ$. O módulo tem uma tensão de entrada de 12 V a 24 V e uma saída de corrente de 4 mA a 20 mA. Relativamente às suas dimensões, apresenta 458,0 mm de altura e 232,0 mm de diâmetro [53]. As principais características destes sensores de vento são apresentados na Tabela 3.3.

3.1.7.2 IMU

Uma unidade de medida inercial IMU é um dispositivo eletrónico que mede e relata a orientação, velocidade e forças gravitacionais recorrendo à combinação de acelerómetros e giroscópios e, frequentemente, magnetómetros [54]. Os IMU são um componente crucial dos sistemas de navegação inercial usados em aeronaves, veículos aéreos não tripulados (*Unmanned Aerial Vehicles (UAV)*) e outros sistemas não tripulados [54]. Os desenvolvimentos recentes permitem a produção de dispositivos GPS habilitados para IMU. Um IMU permite que um receptor de

Tabela 3.3: Características dos sensores de vento estudados.

	PCE-FST-200-201	PCE-FST-200-202	RK120-01C
Tensão de Operação (VDC)	12 - 36	12 - 36	12 - 24
Consumo de Energia (mW)	576 @12V	576 @12V	240 @12V
D x H (mm ²)	127,5 x 131,5	157,5 x 234,0	232,0 x 458,0
Interface de Saída	Tensão e Corrente	Tensão e Corrente	Tensão e Corrente Pulso, RS485, RS232
Velocidade do Vento			
Alcance (m/s)	0,5 - 50	n.a	0 - 45 (Tensão ou Corrente) 0 - 70 (Pulso, RS485, RS232)
Início de medição (m/s)	≤ 0,5	n.a	0,5
Precisão (m/s)	±0,5	n.a	±0,3+0,03V
Direção do Vento			
Alcance (°)	n.a	0 - 360	0 - 360
Início de medição (m/s)	n.a	0,8	0,5
Precisão (°)	n.a	±3	≤ ±3
Resolução (°)	n.a	22,5	1,0
Preço (€)	100 - 200	100 - 200	100 - 200

(n.a. - informação não disponível ou não encontrada pela autora.)

GPS funcione quando os sinais de GPS não estão disponíveis, como em túneis, dentro de edifícios ou quando há interferência eletrônica. As aplicações comuns para IMU incluem controlo e estabilização, navegação e correção, medição e teste, controlo de sistemas não tripulados e mapeamento móvel [54]. Os dados coletados por um IMU são processados por meio de um computador com a finalidade de calcular a posição atual do objeto.

Na Tabela 3.4 são apresentadas as características de alguns módulos inerciais tais como Ellipse 2 Micro Series da SBG Systems [55], MTi 600 Series da XSENS [56], o NGIMU da X-IO Technologies [57], o MPU-6000 da TDK [58] e o BMF055 da Bosch Sensortec [59].

3.1.7.3 GNSS

O GNSS é o termo padrão para sistemas globais de navegação por satélite. Este sensor permite indicar a posição exata do veleiro em qualquer ponto da superfície da Terra. A localização do veleiro é realizada por uma unidade de medida inercial IMU combinada com um recetor GNSS. Usando as acelerações em todos os graus de liberdade e uma alta taxa de atualização, os desvios ocasionais dos dados do GNSS serão suavizados automaticamente. O sistema, portanto, produz um posicionamento altamente preciso [60]. Um recetor GNSS depende de informações recebidas de vários satélites para determinar a posição, velocidade e tempo. Quanto mais sinais de satélite forem capturados pelo recetor, melhor será a precisão de posicionamento [60]. Alguns sensores GNSS foram estudados, nomeadamente o NEO-M8 [61], BN-880 [62] e VN-200 [63]. As principais características destes sensores são apresentadas na Tabela 3.5.

3.1.7.4 Encoder Rotativo

Um *encoder* rotativo é um sensor montado num objeto giratório (como uma roda ou motor) que converte o movimento angular, ou posição, em valores analógicos ou digitais [64]. Este sensor pode ser incremental ou absoluto. O *encoder* absoluto necessita apenas da leitura atual para determinar a posição angular. Em contrapartida, o *encoder* incremental precisa de todas as leituras anteriores para determinar o movimento ou posição angular [64].

No caso do veleiro a desenvolver, decidiu-se recorrer ao *encoder* absoluto de forma a determinar a orientação da vela asa. Alguns exemplos de *encoders* rotativos absolutos são o RM08 desenvolvido pela RLS e o AMT21 da CUIdevices. Na Tabela 3.6 são apresentadas as características desses *encoders*.

3.1.8 Atuadores

O subsistema de atuação é geralmente o consumidor de energia elétrica mais significativo. Segundo Miller *et al.* [65], 75 % dos requisitos de energia apoiam a

Tabela 3.4: Características dos sensores IMU estudados.

	Ellipse 2 Micro	MTi 600	NGIMU	MPU-6000	BMF055
Alcance Acelerómetro (g)	± 16	± 10	± 16	± 16	± 16
Desvios Acelerómetro (mg)	± 5	± 0,01	n.a	± 50	± 80
Alcance Giroscópio (°/s)	± 450	± 2000	± 2000	± 2000	± 2000
Desvio Giroscópio (°/s)	± 0,2	± 0,002	n.a	± 20	± 1
Densidade do ruído da velocidade (m/s/√h)	0,034	0,035	n.a	0,24	0,09
Densidade do ruído angular (°/s/√h)	0,15	0,42	n.a	0,3	0,84
Tensão de Entrada (V)	4,0 - 15,0	4,5 - 24,0	3,1 - 5,0	2,375 - 3,46	2,4 - 3,6
Consumo de Energia (mW)	400	310 - 530	n.a	12,5	50
Interface de Saída	RS-232, RS-422, USB, CAN	XBus, CAN	RS-232, USB	I2C, SPI	SPI
W x L x H (mm ³)	26,8x18,8x9,5	31,5x28x13	50x33x8	4,0x4,0x0,9	1,1x5,2x3,8
Peso (g)	10,0	8,9	10	3,02	n.a
Preço (€)	100 - 200	400-800	200-400	7 - 12	5-15

(n.a. - informação não disponível ou não encontrada pela autora.)

Tabela 3.5: Características dos sensores GNSS estudados.

	NEO-M8	BN-880	VN-200
Precisão da Posição (m)	2	2	1
Canais Recetores	72	72	72
Tipo de Recetores	GLONASS GPS BeiDou Galileo	GLONASS GPS BeiDou Galileo QZSS SBAS	GLONASS GPS
Tensão de Entrada (V)	1,65 - 3,6	2,8 - 6,0	3,2 - 5,5
Consumo de Energia (mW)	63	250	500
Sensibilidade (dBm)	-167	-166	n.a.
Protocolos	NMEA UBX RTCM	NMEA	NMEA
Frequência de Leitura (Hz)	10	10	n.a.
W x L x H (mm ³)	12,2x16,0x2,4	10,0x28,0x28,0	24,0x22,0x3,0
Interface de Saída	UART USB SPI I2C	UART	TTL SPI
Peso (g)	1,6	10,0	4,0
Preço (€)	20 - 40	15 - 30	30 - 70

(n.a. - informação não disponível ou não encontrada pela autora.)

movimentação dos atuadores do sistema para controlar o leme. Por esse motivo, o subsistema de atuação deve ser meticulosamente projetado. Miller *et al.* [65] adotaram atuadores que incorporam um projeto mecânico de parafusos, para que haja pouco ou nenhum consumo de energia quando os atuadores não estão em movimento, mesmo sob carga. Schröder e Hertel [66] apresentaram uma solução de baixo custo que usava motores com uma engrenagem para atingir um alto binário e baixa velocidade, de forma a acionar o leme e navegar.

Com base nas informações do sensor e percepção do ambiente, o veleiro necessita de atuadores, como motores, para controlar a vela asa e o mecanismo de direção. Um servomotor é o ideal visto que permite o controlo preciso da posição angular. Alguns exemplos de servomotores estudados são: o SG90 muito utilizado nos projetos de robótica com o Arduino, Raspberry Pi, entre outros [67], o SER0006 da DFRobot [68], o DSS-M15S [69] e o CS-5226 da Team Corally [70]. As principais características destes servomotores são apresentadas na Tabela 3.7.

Tabela 3.6: Características dos *encoders* rotativos absolutos estudados.

	RM08	AMT21
Resolução (bit)	12	12/14
Precisão (°)	0,3	0,2
Tensão de Entrada (V)	3,3 - 5,0	3,8 - 5,5
Consumo de Energia (mW)	130	80
Velocidade (r/min)	30 000	4000 - 8000
Diâmetro do eixo (mm)	8	9
W x L x H (mm ³)	8,0x3,0x -	28,6x39,9x11,7
Interface de Saída	RS422	RS485
Peso (g)	2,0	15,7
Preço (€)	80 - 90	40 - 50

3.1.9 Unidade de Controlo

A unidade de controlo dos veleiros autónomos geralmente apresenta dois ou três modos de operação distintos: (*i*) controlo de navegação e navegação totalmente autónoma, (*ii*) controlo de vela autónomo com controlo manual de leme e (*iii*) controlo manual completo para iniciar e recuperar o barco e controlar em condições imprevisíveis [71].

De forma ao controlo de navegação e a navegação serem totalmente autónomas, pelo menos três funções básicas devem ser incluídas: (*i*) planeamento global de rotas, (*ii*) prevenção de colisões e (*iii*) o controlo do acompanhamento de rotas [38]. O planeamento global de rotas encontra um caminho ideal sem obstáculos entre os pontos de partida e de destino, com base numa função objetivo e nos dados ambientais e meteorológicos disponíveis. A prevenção de colisões é planeada em tempo real, com possíveis mudanças de rumo devido a obstáculos dinâmicos. Por fim, o controlo de acompanhamento da rota garante que o barco navegue pelo caminho prescrito através do controlo de *feedback* recorrendo ao manuseio da vela e do leme.

O modo autónomo de muitos dos barcos desta pesquisa até agora suportam apenas o acompanhamento de rotas. O planeamento global de rotas geralmente é realizado manualmente, e a prevenção de colisões pode ou não estar presente. Nos veleiros autónomos, o controlador da vela e do leme são habitualmente separados e independentes. A vela é controlada para alcançar a propulsão e a direção é controlada pelo leme. Existem diferentes controladores adotados para controlar os navios e cada um deles traz um conjunto diferente de vantagens e desvantagens entre consumo de energia, poder de computação, facilidade de uso e reconfigura-

Tabela 3.7: Características dos servomotores estudados.

	SG90	SER0006	DSS-M15S	CS-5226
Tensão de Entrada (V)	3,0 - 7,2	4,8 - 6,0	4,8 - 7,2	6,0 - 8,4
Consumo de Energia (W)	0,48 @4,8V / 1,50 @6,0V	2,4	10,8 @6,0V / 14,0 @7,0V	n.a
Binário (kg cm)	1,2 @4,8V / 1,6 @6V	1,6 @ 4,8 V	13,5 @6,0V / 15 @7,0V	11,8 @6,0V / 14,4 @7,4V 16,5 @8,4V
Velocidade Angular (r/min)	83	83	56 @6,0V / 63 @7,0V	143 @6,0V / 167 @7,4V 200 @8,4V
Alcance (°)	0 - 180	0 - 180	0 - 270	0 - 180
W x L x H (mm ³)	12,0x32,0x30,0	12,5x22,0x29,5	20,0x54,5x47,5	20,5x40,5x36
Peso (g)	9	9	n.a	72
Preço (€)	3 - 5	3 - 5	18 - 22	60 - 80

(n.a. - informação não disponível ou não encontrada pela autora.)

bilidade. Para a unidade de controlo principal os dois requisitos primordiais são o baixo consumo de energia e fácil programação. Para obter o baixo consumo de energia é mais adequado um único sistema de microcontrolador, mas isso geralmente ocorre às custas da facilidade de desenvolvimento e teste. Além disso, a modularidade e redundância são outros motivos para considerar mais de um microcontrolador.

Tendo em consideração as condições do vento e a missão do veleiro, o sistema de controlo deverá decidir de forma automática qual o tipo de mareação adequado para atingir o objetivo.

3.1.9.1 Microcontrolador

Os microcontroladores utilizados nos veleiros autónomos têm como objetivo adquirir informações dos diferentes sensores e, subsequentemente, controlar a embarcação operando nos atuadores. Com base nisso, estes devem possuir pinos de *Input/Output* (I/O) suficientes para todos os sensores e atuadores, baixo consumo de energia e periféricos suficientes, tais como, temporizadores, interruptores e interfaces de comunicação, para implementação de todas as funcionalidades. Os microcontroladores STM32 (núcleo-F103RB) incorporam 32 bit ARM®Cortex®-M3 de elevado desempenho operando numa frequência de 72 MHz, assim como uma extensa gama de entradas e saídas e periféricos conetados a dois barramentos. Além disso, este microcontrolador oferece dois *Analog to Digital Converter* (ADC) de 12 bit, três temporizadores de uso geral de 16 bit e possui até duas interfaces de comunicação *Inter-Integrated Circuit* (I2C) e *Serial Peripheral Interface* (SPI), três *Universal Synchronous and Asynchronous Receiver-Transmitter* (USART), um *Universal Serial Bus* (USB) e um *Controller Area Network* (CAN). Estas inúmeras opções de conectividade tornam este microcontrolador ideal para o registo de dados de diversos sensores.

3.1.9.2 Computador de Placa Única

Um *Single-Board Computer* (SBC) é um computador construído num única placa de circuito impresso e contém todos os componentes eletrónicos necessários para o seu funcionamento, incluindo o microprocessador, I/O e memória [72]. Os computadores SBC normalmente fornecem uma solução de baixo consumo de energia para diversas aplicações e um preço baixo. O Raspberry Pi é um SBC que pode executar o sistema operativo Ubuntu Core, que é uma distribuição Linux, necessária para executar o ROS. Alguns exemplos de SBC Raspberry Pi são: o Raspberry Pi 2 [73], Raspberry Pi 3 [73], Raspberry Pi 3 Model B+ [73], Raspberry Pi 3 Model A+ [73], Raspberry Pi 4 Model B 8 GB [73] e Raspberry Pi 400 [73]. As características dos SBC que se consideraram mais relevantes são apresentadas na Tabela 3.8.

Tabela 3.8: Características dos Raspberry Pi estudados.

Raspberry Pi	2	3	3 B+	3 Model A+	4B 8 GB	400
Processador	Cortex-A7	Cortex-A53 64-bit	Cortex-A53 64-bit	Cortex-A53 64-bit	Cortex-A72 (ARM v8) 64-bit	Cortex-A72 (ARM v8) 64-bit
Número de Núcleos	4	4	4	4	4	4
Frequência do CPU	900 MHz	1.2 GHz	1.4 GHz	1.4 GHz	1.5 GHz	1.8 GHz
RAM	1 GB	1 GB	1 GB	512MB	8 GB	4 GB
Interface de Saída	UART, SPI,I2C	UART, SPI, I2C	UART,SPI,I2C	UART,SPI,I2C	UART,SPI,I2C	UART,SPI,I2C
GPIO	40	40	40	40	40	40
Armazenamento	microSD	microSD	microSD	microSD	microSD	microSD
WI-FI	Não	802.11n	2,4GHz and 5 GHz 802.11 b/g/n/ac			
Bluetooth	Não	4.1 BLE	4.1 BLE	4.2, BLE	5.0	5.0
W x H x L (mm ³)	56.5x85.6x17	56.5x85.6x17	56.5x85.6x17	65x56x11	56.5x85.6x11	122x286x26
Consumo de Corrente (A)	0,8	1.35 @ 5 V	1.13 @ 5 V	0.20	1.25 @ 5 V	0.2
Preço (€)	30-45	30 - 45	30 - 45	20-35	60 - 75	55 - 70

3.1.10 Controlo Remoto e Telemetria

Os veleiros autónomos devem operar de forma totalmente autónoma, existindo vários motivos para manter um vínculo de comunicação permanente. Em primeiro, é sempre aconselhável ter um retorno ao controlo manual, no caso de emergências ou para evitar colisões com objetos desconhecidos pelo veleiro. Em segundo, de forma a possuir interesse científico, um robô de navegação com vista a observação oceânica precisa incluir algum sistema de telecomunicações para coligir dados em intervalos regulares, pois não há garantia de que o robô será recuperado. E, para garantir o controlo manual do veleiro (normalmente do leme e das velas), a opção mais comum é utilizar um recetor controlado remotamente.

O veleiro autónomo requer um módulo para comunicação com uma estação base. Esse módulo irá transmitir informações do estado e posição do veículo (telemetria) e receberá, por sua vez, comandos da estação base de forma a controlar o veleiro (controlo remoto).

O *kit* Long Range RF link é um exemplo de módulo a empregar no veleiro desenvolvido pela Seeedstudio [74]. Este dispositivo utiliza uma frequência de rádio de 433,92 MHz com uma largura de banda de 1,55 MHz. Relativamente ao alcance é de 2 km, com sensibilidade de -105 dBm e transmite dados a uma taxa de ± 5 kb/s. Ademais, a tensão de operação é de 3 V - 5 V com um consumo de corrente de 2,5 mA. Em relação às suas dimensões, apresenta um comprimento de 25 mm, uma altura de 7 mm e uma largura de 35 mm [74]. Um outro exemplo de módulo de rádio frequência é o APC220 desenvolvido pela DFRobot [75]. Este dispositivo opera com uma frequência de 420 MHz a 450 MHz e a interface é *Universal Asynchronous Receiver-Transmitter (UART)/Transistor-Transistor Level (TTL)*. O alcance é de 1,2 km e a tensão de operação de 3,5 V - 5,5 V com um consumo de corrente de 25 mA - 35 mA. Relativamente às suas dimensões, apresenta uma altura de 37 mm de comprimento, 17 mm de altura e 6,6 mm de largura [75].

3.1.11 Alimentação do Sistema

Um veleiro autónomo deve ser capaz de navegar sem intervenção humana. Isso quer dizer que todas as tarefas que normalmente são realizadas pelo marinheiro num veleiro convencional devem ser executadas de forma autónoma. O veleiro proposto tem como finalidade navegar durante longos períodos de tempo e, por isso, o sistema de alimentação deve ser robusto e fiável. Por exemplo, pode-se recorrer a um painel solar para fornecer ao veículo energia para alimentar os seus componentes eletrónicos. Este deve ser conetado a uma bateria com saída de energia suficiente para acionar o sistema. No entanto, é fundamental ter em consideração que diferentes tipos de baterias requerem diferentes métodos de carregamento. Ou seja, as baterias de iões-lítio e de chumbo-ácido requerem

uma fonte de tensão constante, em contrapartida as baterias baseadas em níquel necessitam de uma fonte de corrente constante [76].

3.2 Análise e Comparação

Nesta secção é realizada uma análise comparativa dos materiais e componentes elétricos apresentados anteriormente.

3.2.1 Materiais da Estrutura

Relativamente aos materiais anteriormente abordados para o desenvolvimento do casco, a madeira e a fibra de vidro são excelentes opções. Isto é, se o objetivo é desenvolver um veleiro barato de baixo peso, a madeira é o melhor candidato. Por sua vez, se o objetivo é desenvolver um veleiro mais robusto e pesado, a fibra de vidro é a mais apropriada. A mesma analogia surge na escolha dos materiais para a vela e para o mastro.

3.2.2 Casco

Um dos principais objetivos do veleiro proposto é que seja leve, fiável e fácil de controlar. Além disso, tem como intuito navegar durante longos períodos de tempo e decidiu-se que o tipo de casco mais adequado era um mono-casco. Assim sendo, decidiu-se optar por um mono-casco de madeira de tamanho 4,2 m, uma vez que oferece um bom equilíbrio entre custos de construção e recursos de navegação, incluindo a possibilidade de integrar uma maior quantidade de carga útil.

3.2.3 Leme

O veleiro proposto vai apresentar um boca suficientemente larga para permitir o equilíbrio do veleiro, mas não suficiente que justifique o recurso a dois lemes. Por isso, o uso de um leme é suficiente para controlar o veleiro.

Este veleiro tem como propósito ser de fácil controlo e implementação, por isso a simplicidade é um factor muito significativo no desenvolvimento do veleiro.

3.2.4 Vela

Com base na investigação realizada dos diversos veleiros autónomos e nas vantagens e desvantagens das velas tradicionais e das velas rígidas, decidiu-se optar pelo desenvolvimento de um veleiro autónomo com vela rígida. Os principais motivos que conduziram a essa escolha foram os seguintes: (i) as velas rígidas são mais robustas de controlar, sendo o controlo do veleiro o objetivo crucial deste projeto, e (ii) têm melhor eficiência, permitindo ao veleiro navegar com maior

autonomia devido ao menor consumo de energia. Relativamente ao número de velas a empregar no veleiro, considerou-se que o mais apropriado seria um veleiro de vela rígida simples, dado que o controlo é mais simples comparativamente com as velas rígidas duplas. Em relação ao material escolhido para a vela foi a fibra de carbono.

3.2.5 Mastro

No caso do mastro, e com base nas complexidades dos dois tipos de mastro, referidas anteriormente, decidiu-se optar por um mastro de alumínio fixo à base do casco.

3.2.6 Sensor de Vento

Os módulos do sensor de vento apresentados na Tabela 3.3 medem a velocidade do vento, a direção do vento ou ambos. A escolha do módulo determinará se serão necessários um ou dois sensores de vento. Para a elaboração deste projeto, um módulo que possua a velocidade e a direção do vento é o ideal, como por exemplo o sensor RK120-01C, de forma a conseguir-se extrair o máximo de informação do vento num só sensor.

3.2.7 IMU

Os IMU apresentados na Tabela 3.4 apresentam características boas para serem utilizados no veleiro. No entanto, existem diversos IMU com elevados preços. Se o preço não for um factor a ser considerado, a melhor escolha seria o módulo MTi-600 devido à baixa densidade do ruído da velocidade, baixa densidade do ruído angular. Se for pretendida uma mais barata, pode-se optar pelo sensor BMF055 dado que possui uma baixa densidade do ruído da velocidade, baixa densidade do ruído angular. Além disso, possui um magnetómetro integrado, necessário para atualizar a orientação do veleiro.

3.2.8 GNSS

Relativamente aos módulos GNSS apresentados na Tabela 3.5, são todos semelhantes em termos de especificações técnicas e preço. O VN-200 tem a melhor precisão, no entanto a sensibilidade e a taxa de atualização são desconhecidas e apresenta o preço mais alto. O módulo BN-880 apresenta diversas semelhanças comparativamente com o módulo NEO-M8, no entanto apresenta maior peso e maior tensão de entrada. Além disso, apenas possui a interface UART. Com base nessas informações decidiu-se optar pelo sensor GNSS NEO-M8.

3.2.9 Encoder Rotativo

Os *encoders* rotativos absolutos apresentados na Tabela 3.6 determinam, conforme mencionado anteriormente, o valor absoluto da posição angular da vela asa sem a necessidade das leituras anteriores. De forma a determinar a posição angular da vela asa devidamente, é fundamental um *encoder* de boa resolução e precisão, dado que a diferença do ângulo de ataque ser $7,5^\circ$ ou $8,5^\circ$ é significativa nalguns perfis alares. Na escolha de um dos *encoders*, caso o preço seja um fator decisivo, o AMT21 é um codificador ideal e perfeitamente adequado para a tarefa, com um custo inferior ao RM08.

3.2.10 Atuadores

O atuador selecionado dependerá da dimensão do leme assim como dos seus módulos constituintes. Um veleiro exigirá um binário maior no caso do veleiro de vela asa apresentar uma envergadura maior ou um tipo diferente de material que conduzirá a um peso maior. Assim como, um veleiro com dimensão maior do leme exigirá um servomotor de maior binário. Na Tabela 3.7 são apresentados diversos servomotores de menor e maior binário. Além disso, tendo por base que os servomotores de maior binário apresentam maior consumo de energia e peso do que os servomotores de menor binário, deve-se escolher o servomotor com o menor binário necessário de forma a ser o mais eficiente possível em termos energéticos. Dessa forma, o servomotor escolhido para controlar o mecanismo de direção do veleiro é o DSS-M15S [69]. E, o servomotor escolhido para controlar a cauda é o SER0006 [68].

3.2.11 Microcontrolador

Com base nas características do microcontrolador STM32 abordado anteriormente, concluiu-se que é um dispositivo ideal devido aos seus vários periféricos e interfaces de comunicação. Este microcontrolador possui I/O suficientes para cada sensor e atuador que se pretende empregar no veleiro.

3.2.12 Computador de Placa Única

O Raspberry Pi é um SBC que pode executar o sistema operativo Ubuntu Core necessário para executar o ROS. É, portanto, fundamental ter uma distribuição Linux numa máquina de 64 bit. E, devido a esse factor, o Raspberry Pi 2 pode ser excluído. Os restantes SBC, ilustrados na Tabela 3.8, apresentam preços muito competitivos e são todos boas opções. No entanto, o Raspberry Pi 4B 8 GB possui uma *Central Processing Unit* (CPU) ligeiramente melhor e uma *Random-Access Memory* (RAM) superior.

3.2.13 Controlo Remoto e Telemetria

O veleiro, ao longo da sua missão, tem de comunicar com a estação base relativamente à sua localização e outras informações que se possam considerar interessantes partilhar. Com isto, decidiu-se que a comunicação do veleiro será via rádio. Este tipo de comunicação é muito utilizada quando se pretende executar contactos à distância entre duas estações.

Uma estação de radiocomunicação é um conjunto de equipamentos necessários para a comunicação via ondas de rádio. Geralmente, é um receptor ou transmissor ou transceptor, uma antena e algum equipamento adicional necessário para operá-los. Eles desempenham um papel vital na tecnologia de comunicação, pois são fortemente dependentes da transferência de dados e informações em todo o mundo. Dessa forma, para o controlo remoto e telemetria, optou-se pelo módulo de radiocomunicação APC220 [75].

3.2.14 Alimentação

A alimentação do veleiro provém do painel solar e da bateria responsável por alimentar toda a componente eletrónica.

Na Tabela 3.9 é apresentado o consumo estimado de potência do sistema proposto, totalizando 33,833 W, tendo por base as informações dos componentes apresentados anteriormente.

Tabela 3.9: Potência consumida do sistema.

Componentes	Potência Consumida (W)
RK120-01C	0,240
BMF055	0,050
NEO-M8	0,063
AMT21	0,080
DSS-M15S	14,000
SER0006	2,400
Raspberry Pi 4B 8 GB	15,000
APC220	2,000
Total	33,833

Para a alimentação, uma bateria *Lithium Polymer* (LiPo) com uma capacidade de 22,2 V de 4400 mA h ($22,2 \text{ V} \times 4,4 \text{ A h} = 97,68 \text{ W h}$) e uma taxa de descarga de 45 C ($4,4 \text{ A/h} \times 45 \text{ C} = 198 \text{ A}$) é proposta. No caso de todos os componentes estarem a consumir o máximo de energia e se não houver energia

gerada pelo painel solar, a autonomia da bateria é de 2,88 h. Além disso, é utilizado também um conversor *buck-boost*, com a particularidade de obter tensões de saída com um valor maior ou menor do que a entrada, para estabelecer conexão entre o painel solar e a bateria, tornando a tensão de entrada da bateria constante.

3.3 Conclusão do Capítulo

Neste capítulo inventariaram-se e selecionaram-se diferentes componentes necessários para desenvolver um veleiro autónomo equipado com uma vela rígida como propulsão. Devido à pandemia COVID-19 surgiu a necessidade de implementar o projeto na simulação, e não na construção. No entanto, foram efetuados estudos acerca dos diversos constituintes do veleiro.

A seleção dos materiais que constituem o casco, a vela e o mastro foram dos primeiros estudos realizados. De seguida, foi apresentado duas possíveis soluções de casco, mono-casco ou multi-casco e as suas vantagens, sendo a seleção escolhida o mono-casco. Relativamente ao leme, optou-se por um leme simples uma vez que permite um controlo mais simples. E, o patilhão pretendeu-se com alongamento elevado por questões de estabilidade. Em relação à vela, realizou-se um estudo entre velas rígidas simples e velas rígidas duplas, apresentando as suas vantagens e desvantagens. E, a seleção escolhida foi a vela rígida simples de fibra de carbono, uma vez que o controlo é mais simples comparativamente com as velas rígidas duplas e a fibra de carbono apresenta uma densidade leve. O mastro selecionado é de alumínio fixo à base do casco. Relativamente aos sensores selecionados, optou-se por um sensor de vento que possua a velocidade e a direção do vento, um sensor IMU com baixa densidade do ruído da velocidade e baixa densidade do ruído angular e um sensor GNSS de baixo custo com diversas interfaces de saída, comparativamente com os restantes sensores estudados. O *encoder* rotativo selecionado foi o *encoder* absoluto uma vez que necessita apenas da leitura atual para determinar a posição angular, ao contrário do *encoder* incremental que precisa de todas as leituras anteriores para determinar o movimento ou a posição angular. Em relação aos atuadores decidiu-se optar pelos servomotores com o menor binário necessário, de forma a ser o mais eficiente possível em termos energéticos. O microcontrolador escolhido foi o STM32 uma vez que possui vários periféricos e I/O suficientes para cada sensor e atuador. O SBC selecionado foi o Raspberry Pi porque permite executar o sistema operativo Ubuntu Core necessário para executar o ROS. Relativamente ao controlo remoto e telemetria, optou-se pelo módulo de radiocomunicação. Por fim, relativamente à alimentação optou-se por uma bateria de LiPo.

Em suma, concluiu-se que os componentes selecionados devem consumir o mínimo de energia possível.

Capítulo 4

Proposta de Modelação e Controlo do Veleiro

Este capítulo propõe uma solução para veleiros autónomos com base no estudo apresentado anteriormente. Além disso, apresenta o perfil alar selecionado e o funcionamento da vela asa com cauda.

Uma das exigências desta dissertação é adotar a estrutura ROS para o controlo do veleiro. Além disso, é necessário analisar um conjunto de características para conceber e modelar a plataforma. Contudo, devido à pandemia COVID-19, a dissertação concentra-se na simulação, e não na construção, montagem e teste da plataforma projetada.

4.1 Proposta

Nesta secção é proposto um veleiro com vela asa conforme as necessidades apresentadas anteriormente. De forma a que o veleiro seja robusto, é proposto para o casco uma estrutura de madeira, com um comprimento de 4,2 m. Dada a densidade da madeira 485 kg/m^3 , o casco apresenta uma massa de 150 kg. No que diz respeito à vela asa apresenta uma perfil alar simétrico com cauda. A cauda possui o mesmo perfil alar e o seu objetivo é definir o ângulo de ataque da vela asa. A carga útil máxima prevista, incluindo a vela asa, a cauda e todos os componentes eletrónicos, é de aproximadamente 40 kg. Em relação ao sistema de controlo, o servomotor proposto é o DSS-M15S [69], com um binário de 13,5 kg/cm, para controlar o mecanismo de direção do veleiro com massa total de 190 kg. De forma a controlar a cauda é proposto um servomotor SER0006 [68], com um binário de 1,6 kg/cm, dado que o controlo da cauda exigirá menos binário do que o necessário para controlar a direção do veleiro. O veleiro possui o sensor inercial

BMF055 [59] e uma unidade NEO-M8 [61] GNSS para determinar a sua orientação e posição. Em relação à velocidade e direção do vento aparente recorreu-se ao sensor RK120-01C [53]. No que diz respeito ao SBC considerou-se o Raspberry Pi 4B 8 GB o mais adequado para controlar o sistema.

Na Figura 4.1 pode-se observar o veleiro proposto e na Figura 4.2 um diagrama de blocos da arquitetura do sistema, onde as áreas azul, lilás e verde correspondem à alimentação do sistema, aos sensores e aos atuadores, respetivamente.

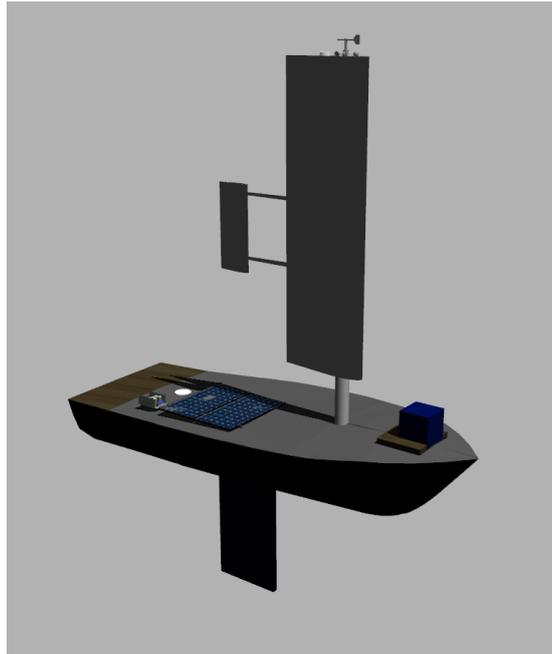


Figura 4.1: Veleiro proposto.

4.2 Vela Asa

A vela rígida propulsiona o barco através da força de sustentação (*lift*) gerada quando o fluxo de ar interage com a vela. A força depende da secção transversal da vela, ou seja, do perfil alar [13]. A sustentação gerada por um perfil alar é o movimento relativo entre o fluido e o perfil alar resultando no ângulo de ataque (α), definido como o ângulo entre a corda média da vela e a direção do vento aparente (W_a). Quando orientado num ângulo adequado, o perfil alar desvia o ar que se aproxima, resultando numa força sobre o perfil alar na direção oposta à deformação. Essa força é denominada como força aerodinâmica, sendo dividida em duas componentes [13]:

- a componente perpendicular à direção do vento aparente, o *Lift* (L) ou

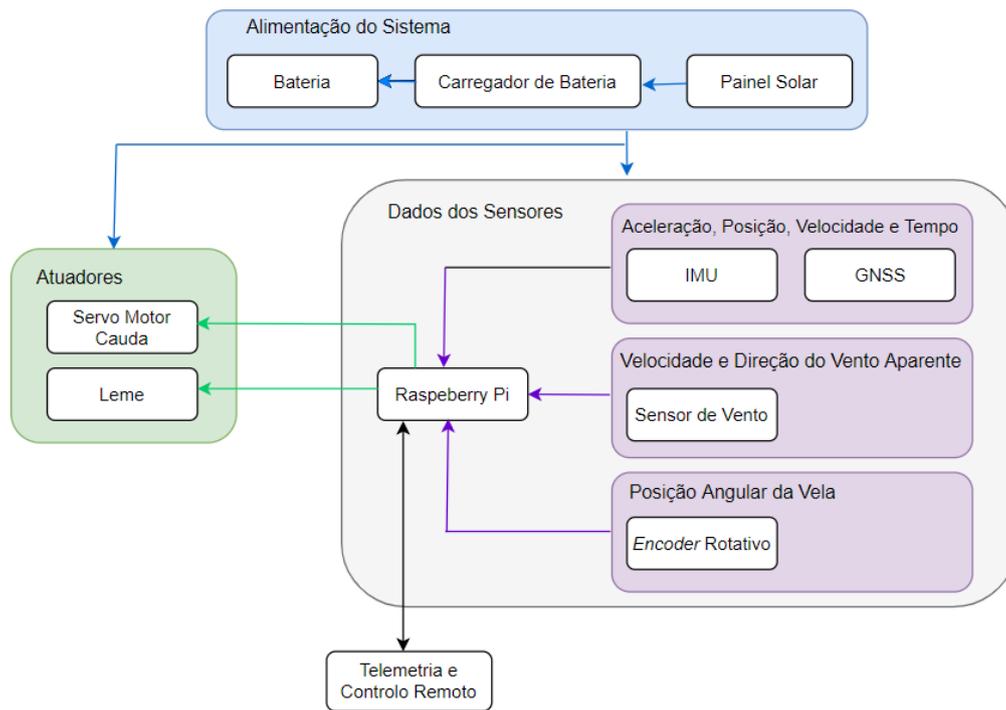


Figura 4.2: Arquitetura do sistema.

sustentação;

- a componente paralela à direção do vento, o *Drag* (D) ou resistência aerodinâmica.

Os perfis alares simétricos requerem um ângulo de ataque positivo para gerar sustentação. Contudo, os perfis alares assimétricos podem gerar sustentação com um ângulo de ataque zero. Na Figura 4.3 pode-se observar as forças aerodinâmicas e o ângulo de ataque envolvido no perfil alar.

A sustentação e a resistência aerodinâmica produzidos por um perfil alar são dados pela Equação 4.1 e Equação 4.2, onde ρ representa a densidade do ar, V

$$L = \frac{1}{2}\rho V^2 AC_L \quad (4.1)$$

$$D = \frac{1}{2}\rho V^2 AC_D \quad (4.2)$$

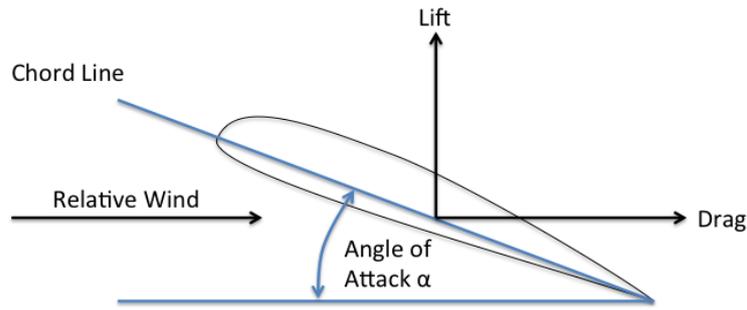


Figura 4.3: Forças aerodinâmicas e ângulo de ataque α no perfil alar [11].

a velocidade da vela rígida, A a superfície da vela rígida, C_L o coeficiente de sustentação do perfil alar e C_D o coeficiente da resistência aerodinâmica do perfil alar.

4.2.1 Seleção do Perfil Alar

O veleiro, ao longo da sua trajetória, está sujeito a diversas condições atmosféricas, nomeadamente, ventos fortes. Para que a vela se comporte sempre da mesma forma, independentemente da direção do vento, deve-se escolher um perfil alar com secção simétrica, apesar dos perfis alares assimétricos apresentarem normalmente um coeficiente de sustentação (C_L) máximo mais alto.

O Número de Reynolds (R_e), ilustrado pela Equação 4.3, representa a razão das forças cinemáticas ou inerciais e a força da viscosidade de um fluido, onde ρ representa a densidade do meio, V a velocidade do fluxo, L o comprimento característico e μ a viscosidade do meio [13].

$$R_e = \frac{\rho V L}{\mu} \quad (4.3)$$

De forma a escolher o perfil alar adequado da vela rígida, foram analisadas as principais características de diversos perfis alares simétricos compreendidos entre $200\,000 < R_e < 500\,000$, de acordo com a análise elaborada por Silva *et al.* [13]. O desafio foi encontrar um perfil alar que exibisse valores elevados de C_L/C_D , C_{Lmax} , assim como uma espessura máxima elevada para melhorar a rigidez da estrutura da vela. Com base nestes critérios, elegeu-se o perfil alar EPPLER 479, ilustrado na Figura 4.4.

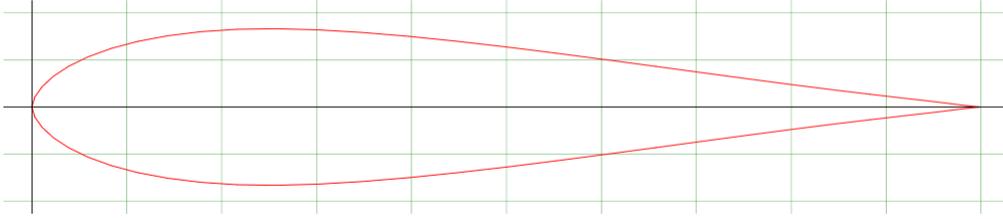


Figura 4.4: Secção do perfil alar EPPLER 479 [12].

As principais características aerodinâmicas do perfil alar EPPLER 479, para $R_e = 200\,000$, estão representadas nos gráficos da Figura 4.5. Com base nestes gráficos, conclui-se que este perfil alar apresenta valor de $C_L/C_D \approx 50$ para ângulos de ataque no intervalo $5^\circ < \alpha < 10^\circ$. Nesse intervalo de α , o coeficiente de arrasto, $C_D < 0,02$. Para $\alpha \approx 10^\circ$ o valor de C_L é de $\approx 1,1$.

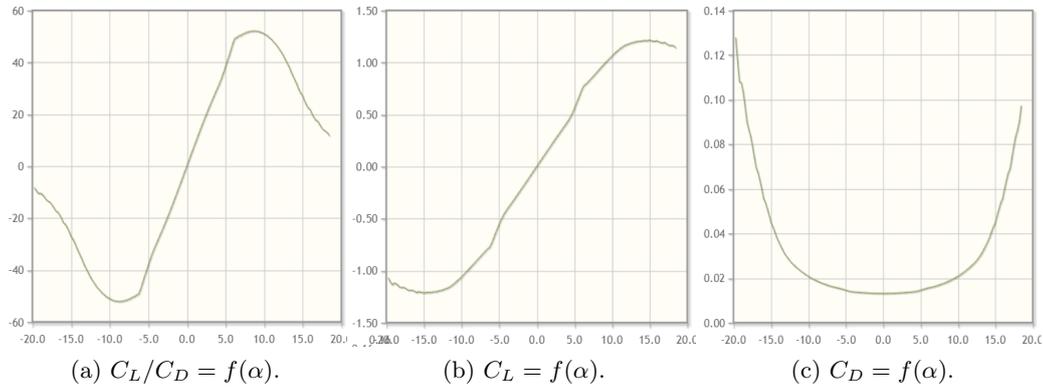


Figura 4.5: Principais características do perfil alar Eppler 479 [12].

4.2.2 Forças Aplicadas

A força necessária para propulsionar o veleiro é dada pela segunda lei de Newton apresentada na Equação 4.4, onde m e a são a massa e a aceleração do veleiro, respectivamente. De forma a que o veleiro inicie o seu movimento, a aceleração inicial a não deve ser nula ($|a| > 0$). Neste caso, a massa do veículo, que compreende o casco, conjunto de vela asa e cauda e carga útil máxima, é considerada como sendo 190 kg.

$$F = ma \quad (4.4)$$

As principais forças envolvidas numa vela rígida quando o veleiro navega

contra e a favor do vento estão ilustradas na Figura 4.6. Onde F é a força aerodinâmica que se decompõem em H , componente de adorno, e T , a força de propulsão que efetivamente impulsiona o veleiro, L e D são as forças de sustentação e arrasto, respetivamente e α é o ângulo de ataque [13]. Consequentemente, a força necessária para propulsionar o veleiro é dada pela Equação 4.5, onde β é o ângulo entre o vetor de sustentação e o rumo do veículo, m é a massa do veleiro, a é a aceleração:

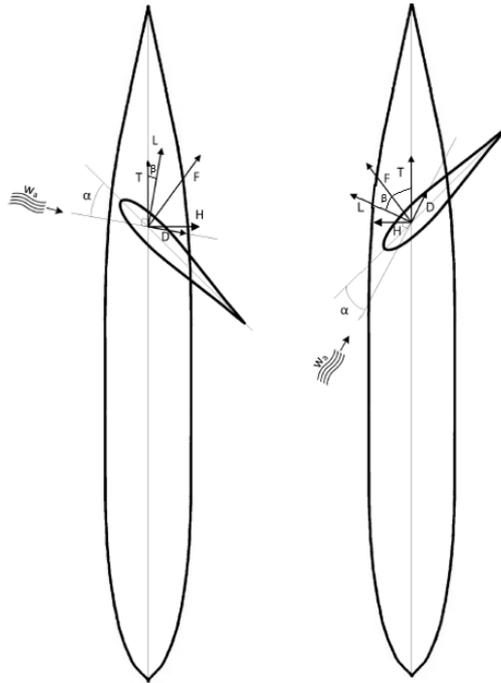


Figura 4.6: Forças na vela rígida ao navegar contra o vento (esquerda) e a favor do vento (direita) [13].

$$L \cos \beta = ma + F_{at} \quad (4.5)$$

Substituindo o fator de sustentação na Equação 4.5 pela expressão de sustentação na Equação 4.1 e dado que a força de atrito pode ser expressa pela Equação 4.6:

$$F_{at} = \mu mg \quad (4.6)$$

Onde μ é o coeficiente de atrito, m é a massa do veleiro e g é a aceleração gravitacional, a Equação 4.7 é obtida da seguinte forma:

$$\frac{1}{2}\rho V^2 AC_L \cos\beta = ma + \mu mg \quad (4.7)$$

O vetor de sustentação é perpendicular à direção aparente do vento e o vetor T é paralelo ao eixo x do veleiro, o que significa que β pode ser expresso pela Equação 4.8:

$$\beta = |W_a + 90| \quad (4.8)$$

Onde W_a é a direção do vento aparente. A vela asa pode gerar uma força de sustentação efetiva com direções aparentes do vento de $35^\circ \leq W_a \leq 135^\circ$ e $-35^\circ \geq W_a \geq -135^\circ$.

4.2.3 Ajuste da Vela Asa

As estratégias de controlo da vela em veleiros autónomos baseiam-se essencialmente no controlo do ângulo de ataque da vela rígida (o ângulo da vela em relação ao vento) [13]. O controlo depende da forma como a vela asa é inserida no veleiro: (i) a vela é fixada diretamente ao mastro, movendo-se com a rotação do mastro; ou (ii) a vela gira livremente em torno do mastro, funcionando como um cata-vento. No caso (ii) existem duas possibilidades para accionar a vela rígida e controlar o ângulo de ataque. A primeira possibilidade é controlar o ângulo de ataque mecanicamente, recorrendo a um atuador que gire a vela rígida em torno do mastro [28]. Este método possui a vantagem de um acionamento rápido e posicionamento correto do eixo. Todavia, a variabilidade do vento exige atuação de alta frequência para manter a vela rígida corretamente ajustada. A segunda possibilidade para o controlo do ângulo de ataque é o recurso a uma superfície auxiliar para compensar a vela rígida aerodinamicamente [28]. A superfície auxiliar pode assumir a forma de uma cauda atrás da vela rígida (convencional) ou uma cauda na frente da vela rígida (*canard*), com perfil idêntico à vela rígida, permitindo simplificar o controlo assim como a vela alcançar automaticamente o ângulo ideal em relação ao vento [13]. Nestas circunstâncias, as velas rígidas designam-se de auto-ajustáveis. Além disso, esta configuração permite ajustar o ângulo de ataque da vela em relação ao vento, de acordo com a velocidade do vento e, assim, variar a força resultante [13]. Esta opção (ii) apresenta maior fiabilidade, controlo mais fácil e menor consumo de energia.

4.2.3.1 Asa com Canard

Uma asa *canard* é um arranjo de asa que fica na frente da vela asa principal. Este arranjo apresenta uma distribuição de peso bastante equilibrada e, por isso, requer menos lastro para posicionar o centro de massa da vela asa na posição desejada, em comparação com o arranjo de asa com cauda [13].

4.2.3.2 Asa com Cauda

Com o intuito de melhorar as características do perfil aerodinâmico simétrico, algumas velas rígidas incluem uma cauda com o intuito de criar uma sustentação adicional, como também auxiliar a vela principal a atingir a sua capacidade máxima de sustentação [13].

O princípio de funcionamento de uma vela rígida controlada por cauda é relativamente simples. Normalmente, a direção do vento aparente em relação à vela asa é determinada por um cata-vento ou um anemómetro colocado no topo do mastro e o ângulo de ataque da vela asa resulta da diferença entre a direção da vela asa e a direção do vento aparente.

Quando a cauda e a vela asa estão alinhadas a vela asa comporta-se como um cata-vento, apresentando um ângulo de ataque nulo – Figura 4.7 (parte superior) [13]. Para imprimir um ângulo de ataque à vela, atua-se sobre a cauda – Figura 4.7 (centro). Esta ação cria um binário em torno do mastro da vela principal, levando a vela a girar na direção oposta à da cauda para equilibrar o binário do sistema vela-cauda – Figura 4.7 (parte inferior). A vela asa passa a apresentar um ângulo de ataque positivo em relação ao vento aparente, gerando sustentação e arrasto. A componente de sustentação permite acelerar o veleiro até que o arrasto no casco seja igual à propulsão da vela. Mantendo-se estas condições, o veleiro movimenta-se a uma velocidade constante [13].

O momento de equilíbrio entre a asa e a cauda mantém o ângulo de ataque da vela constante em relação ao vento, propulsionando o barco. Assim, caso o vento mude de direção, o conjunto asa-cauda acompanha a mudança de direção. A sustentação só pode ser gerada perpendicularmente ao vento relativo, por isso, o vetor de sustentação também gira para a mesma posição em relação ao vento. Dessa forma, toda a vela asa, cauda e vetor de sustentação giram junto com o vento como uma unidade rígida [13]. No entanto, caso o vento cruze a linha central do barco, a posição do *flap* e da cauda devem ser invertidas, o que corresponde à virada por davante ou virada em roda, dependendo se o vento cruza a linha central voltado para a proa ou para a popa, respetivamente [13].

De forma a usar a vela asa como sistema de propulsão do veleiro, esta precisa de cumprir um conjunto de requisitos. Para colocar a vela asa numa posição estável e obter a propulsão desejada, o momento em torno do eixo de rotação

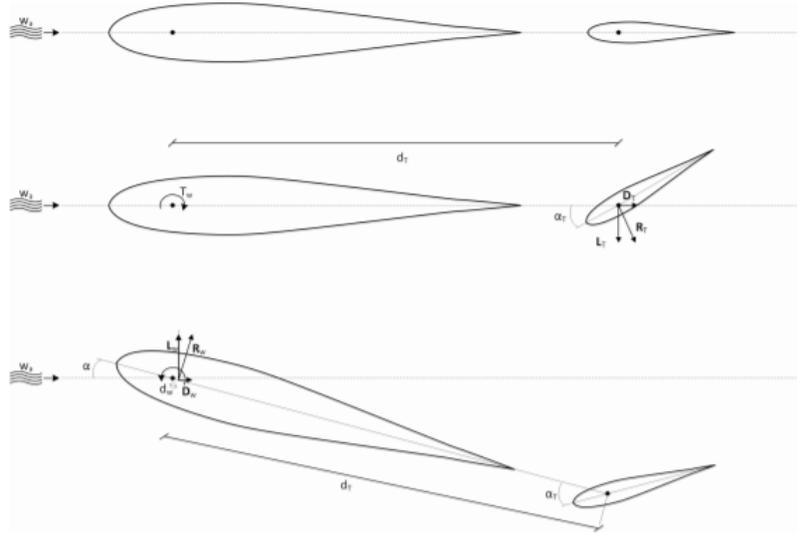


Figura 4.7: Forças e binários envolvidos no controlo da vela: cauda não acionada (parte superior), cauda acionada (centro) e vela asa e cauda em equilíbrio (parte inferior) [13].

da vela asa (eixo z) precisa ser zero, caso contrário a vela gira. Esse estado de equilíbrio é definido pela Equação 4.9, onde T_W representa o momento de toda a asa.

$$T_W = 0 \quad (4.9)$$

O principal critério a ser considerado é a estabilidade do ângulo de ataque em relação à mudança na direção ou velocidade do vento [28].

Com base no perfil alar selecionado anteriormente, pretende-se especificar as características da vela asa e da cauda. Segundo Elkaim [28], realisticamente, o coeficiente de sustentação operacional máximo para a vela tradicional é de 0,6. Para o perfil alar escolhido, o coeficiente de sustentação máximo para um ângulo de ataque $\alpha \approx 10^\circ$ é de $C_{Lmax} \approx 1,1$. Na equação 4.10 calculou-se a razão entre o coeficiente de sustentação do perfil alar escolhido, C_{LE479} , para o ângulo de ataque $\alpha \approx 10^\circ$ e o coeficiente de sustentação máxima para a vela tradicional, C_{Lw} :

A razão indica que a área vélica da vela rígida deve ser 1,83 vezes menor que a vela tradicional, o que corresponde a uma área vélica de $3,8 \text{ m}^2$ dado que, como dito anteriormente, a área da vela tradicional do casco Proton 2,4 mR é de 7 m^2 .

$$C_{Lr} = C_{LE479}/C_{Lw} \approx 1,83 \quad (4.10)$$

Porém, como a relação C_L/C_D do perfil alar Eppler 479 em torno de $\alpha \approx 10^\circ$ é melhor em comparação com a vela tradicional, decidiu-se que a vela rígida deveria ter uma área de $A_w \approx 1,88 \text{ m}^2$. Tendo como base a mesma analogia, a superfície auxiliar apresenta uma área de $A_t \approx 0,23 \text{ m}^2$.

Relativamente ao formato geométrico da vela asa foram consideradas duas formas [13]: (i) uma vela asa retangular com uma altura de 2,50 m e uma largura de 0,75 m, e (ii) uma vela asa trapezoidal com uma altura de 2,50 m e uma largura a variar entre 0,55 m a 0,95 m. O formato da vela trapezoidal apresenta vantagem relativamente ao centro de pressão aerodinâmico, sendo mais baixo para o mesmo valor de sustentação gerado. Apesar disso, decidiu-se optar pela vela de asa retangular devido à sua simplicidade de construção em comparação com a vela trapezoidal. A cauda apresenta também um formato retangular com uma envergadura de 0,75 m e um comprimento de corda de 0,30 m. Na Tabela 4.1 pode-se observar as dimensões da vela asa e cauda.

Tabela 4.1: Dimensões da vela asa e cauda.

	Envergadura (m)	Corda (m)	Área (m ²)	Perfil Alar
Vela Asa	2,50	0,75	1,88	EPPLER 479
Cauda	0,75	0,30	0,23	EPPLER 479

4.3 Casco

Esta secção apresenta brevemente as diversas opções de casco consideradas e o casco selecionado.

Existem diversos projetos possíveis de casco para barcos à vela autónomos. Miller *et al.* [77] apresentam uma série de estudos de desempenho sobre os recursos de projeto do casco e os correspondentes efeitos de desempenho. Um casco ideal seria barato de fabricar, pequeno o suficiente para facilitar o transporte e evitar os danos relacionados à colisão, mas grande o suficiente para poder navegar efetivamente em marés inconstantes [77]. Miller *et al.* [65] comparam o mono-casco com multi-casco e afirmam que catamarãs e trimarãs demonstraram um desempenho significativamente superior ao mono-casco em muitas aplicações que

não incluem grandes mudanças no deslocamento. Por outro lado, eles identificam dois desafios com multi-casco: (i) relativamente pesados e (ii) normalmente não são auto-endireitáveis e, portanto, podem ser difíceis de recuperar de um emborcamento [65].

No que diz respeito ao mono-casco existem três possibilidades principais para o comprimento total do casco: (i) um casco destinado a barcos-modelo controlados por rádio (com menos de 2 m de comprimento), (ii) um casco pequeno (3 m a 5 m de comprimento) e (iii) modificar um casco do tamanho de um iate (com mais de 5 m de comprimento). Para as diferentes possibilidades de mono-casco existem as suas vantagens. A embarcações mais curtas, opção (i), são baratas, leves, fáceis de manusear e simples de construir [13]. Neste tipo de embarcação, os testes podem ser facilmente efetuados sem a necessidade de qualquer infraestrutura especial, como uma rampa de deslizamento. Por outro lado, um barco desse tamanho é profundamente sensível mesmo a pequenas rajadas de vento e ondas. Além do mais, o seu espaço restrito impede equipamentos adicionais. Os barcos com casco conforme a opção (iii) apresentam características opostas à da opção (i). Nos últimos anos, a opção (ii) parece estar a ganhar força tanto para projetos académicos, quanto para produtos comerciais [13].

Face às considerações acima apresentadas, optou-se por um mono-casco de tamanho “médio”. E, após uma pesquisa de possíveis cascos existentes no mercado, dadas as dimensões, robustez de construção, elevada estabilidade e sistema simples de leme, optou-se por um casco Proton 2,4 mR [14] ilustrado na Figura 4.8, com comprimento total de 4,16 m. Este tipo de casco é usado para as Para World Sailing Championships, que envolvem pessoas com mobilidade reduzida, devido ao facto de apresentarem uma elevada estabilidade.



Figura 4.8: Casco Proton [14].

O Proton 2,4 mR foi projetado e fabricado de acordo com as regras “*International 2.4 One Design Class Rules*”. Este veleiro é entregue com o “Certificado

de 2,4 mR Classe Iate” e a placa de construção da International Sailing Federation (ISAF) que mais tarde, em 2015, passou a ser chamado de World Sailing, permitindo participar em regatas internacionais [14].

O Proton foi desenvolvido e preparado para navegar em condições climáticas adversas, sendo construído em fibra de plástico. O assento do timoneiro é ajustável, e todos os cabos são acessíveis e controláveis com facilidade e segurança. Neste veleiro, o sistema de controlo do leme é simples e duplo: pedais ou leme manual localizado na frente do timoneiro [14]. Os pedais do leme possuem um sistema de ajuste duplo, onde o ângulos dos pedais e a distância do assento do timoneiro são ajustáveis [14]. No que se refere à estabilidade, além de todos os componentes que contribuem para o centro de massa do veleiro, as pessoas também colaboram. Por estas razões, o 2,4 mR é uma embarcação usada nas competições de vela adaptada praticadas por pessoas de mobilidade reduzida.

Esta embarcação, na sua versão com vela tradicional, apresenta uma área total vélica de 7 m^2 e um comprimento de mastro de 4,65 m [14]. No entanto, no caso deste projeto com vela rígida, o mastro é de alumínio e apresenta um comprimento de 3,0 m (altura desde a cobertura do casco).

4.4 Sistema de Controlo

O sistema de controlo é responsável pela posição angular da vela na posição desejada, por meio da cauda, assim como acionar o mecanismo de direção de forma a que o veleiro mantenha a trajetória. Dessa forma, existem dois sistemas de controlo fundamentais, o controlo da vela asa e o controlo da direção do veleiro.

4.4.1 Controlo da Vela Asa

Para controlar a posição angular da vela asa, o sistema requer como entradas a direção do vento, a posição angular da vela asa atual e o ângulo de ataque desejado. Primeiro, o ângulo de ataque atual é calculado a partir da direção do vento aparente e da direção angular da vela asa. O ângulo de ataque atual é subtraído ao ângulo de ataque desejado e a diferença é fornecida como uma entrada para o controlador. Finalmente, o controlador determina a posição angular ótima da cauda e envia o sinal de *Pulse Width Modulation* (PWM) necessário para o servomotor da cauda, conforme mostrado na Figura 4.9.

4.4.2 Controlo da Direção do Veleiro

Com o intuito de controlar a posição e orientação do veleiro, este deve possuir um sistema de controlo de direção. A orientação e posição atual do veleiro são obtidas com base nos dispositivos IMU e GNSS, respetivamente. Estas informações são subtraídas à orientação e posição desejadas do veleiro, de forma a calcular

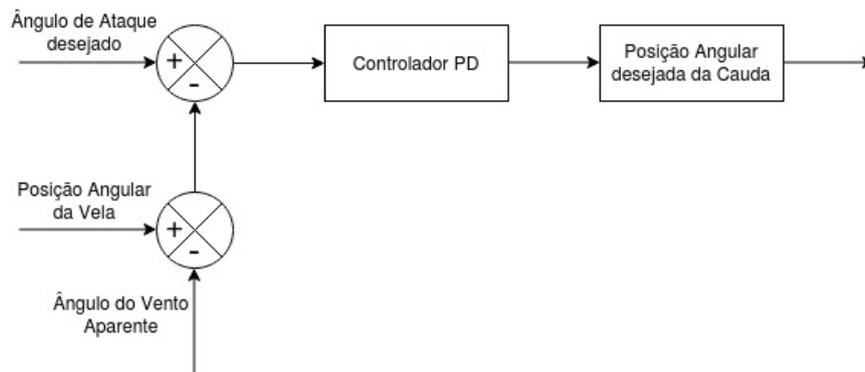


Figura 4.9: Sistema de controlo da cauda.

o ângulo de direção desejado. De seguida, o controlador gera o sinal PWM correspondente para o controlo de direção. A Figura 4.10 apresenta o diagrama do sistema de controlo de direção proposto.

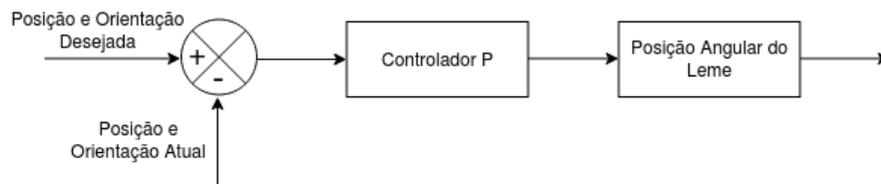


Figura 4.10: Sistema de controlo do leme.

4.5 Conclusão do Capítulo

Neste capítulo foi elaborada uma proposta de solução para o veleiro, tendo em conta as características identificadas no Capítulo 3.

Relativamente ao arranjo de asa concluiu-se que o arranjo de asa *canard* apresenta uma distribuição de peso mais equilibrada em comparação com o arranjo de asa com cauda. Isto significa que uma asa *canard* requer menos lastro para posicionar o centro de massa da vela asa na posição desejada. Todavia, em condições variáveis de vento, o arranjo de asa *canard* quando livremente rotacional, é mais instável do que a vela asa com cauda [13]. Dessa forma, decidiu-se recorrer à cauda como superfície auxiliar para a vela asa. O perfil alar da vela asa e da cauda escolhido foi o EPPLER 479.

Em relação ao casco optou-se por um mono-casco Proton 2,4mR com comprimento de 4,2m, devido ao facto de apresentar uma elevada estabilidade.

Neste capítulo também foi abordado o sistema de controlo da vela asa e da direção do veleiro, optando por um controlador PD e P, respetivamente.

Capítulo 5

Ferramentas de Simulação

Neste capítulo vão-se abordar as ferramentas de simulação de dinâmica de fluídos compatíveis com o ROS, que permitem a simulação do comportamento de veleiros. Os ambientes de simulação que se consideraram foram: o UWSim, CoppeliaSim e Kelpie. Os simuladores abordados foram: o Modular Open Robots Simulation Engine (MORSE), Gazebo e USVSim. Além disso, é efetuada uma análise comparativa destas ferramentas de simulação de forma a selecionar a mais adequada para a realização deste projeto.

5.1 Ambientes de Simulação

Cada vez mais, a simulação robótica é uma ferramenta essencial, dado que um simulador bem projetado torna possível analisar os algoritmos, projetar robôs, executar testes em ambientes realistas, entre outros, com o objetivo de averiguar o ponto de situação do projeto.

5.1.1 UWSim

O UWSim é uma ferramenta modular e extensível para visualização e simulação de missões robóticas subaquáticas. O UWSim oferece simulações de ondas visualmente realistas, funcionando essencialmente como uma ferramenta de visualização de módulos externos, responsável por algoritmos de controlo, veículos e dinâmica do ambiente [78].

Este simulador visualiza um cenário virtual subaquático. Os veículos submarinos controláveis, embarcações de superfície e manipuladores robóticos, bem como sensores simulados, podem ser adicionados ao cenário virtual e acedidos externa-

mente por meio da interface ROS, permitindo integrar facilmente as ferramentas de visualização com arquiteturas de controlo existentes [79].

A Figura 5.1 apresenta as principais componentes e classes do UWSim. O UWSim possui [15]: (i) um módulo *Core*, responsável por carregar o cenário principal e os seus robôs simulados, (ii) um módulo *Interfaces* que fornece comunicação com arquiteturas externas, (iii) um módulo *Dynamics* que implementa a dinâmica do veículo subaquático, (iv) um módulo de *Physics* que gere os contactos entre os objetos no cenário, (v) o *osgOcean* responsável por renderizar a superfície do oceano, e (vi) o módulo *Graphical User Interface* (GUI) que fornece suporte para visualização.

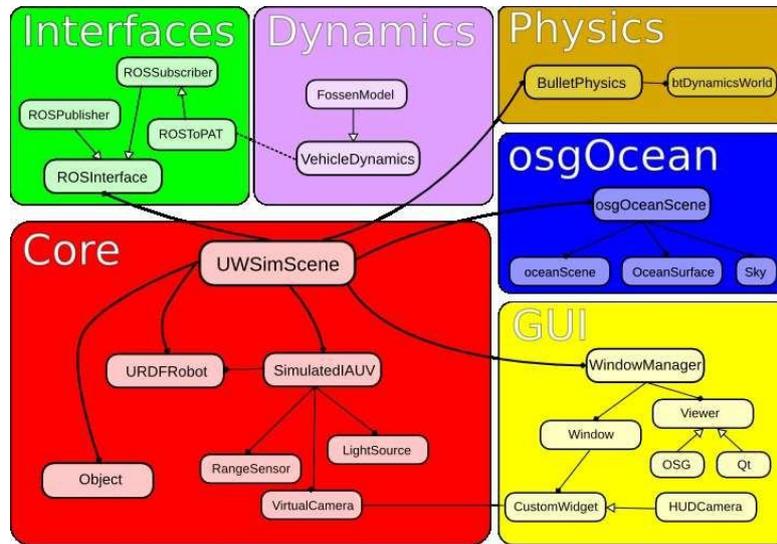


Figura 5.1: As principais partes que constituem o UWSim [15].

As principais limitações do UWSim para simulação do barco são: (i) a falta de dinâmica do leme, (ii) os modelos simplificados para a simulação da corrente marítima, (iii) a simulação do vento parecer afetar apenas a forma e a altura da onda, não o movimento do veículo, e (iv) a dificuldade em modelar o comportamento realista do barco, dado que o foco deste simulador é nos *Unmanned Underwater Vehicles* (UUV) [15].

5.1.2 CoppeliaSim

O CoppeliaSim é um simulador robótico desenvolvido pela Coppelia Robotics. Este simulador era conhecido no passado como VREP. Este *software* oferece a possibilidade de controlar cada modelo ou objeto por meio de uma *Application Programming Interface* (API), um *plugin*, um nó ROS, entre outros. Isto torna o CoppeliaSim um simulador versátil e ideal para aplicações robóticas.

O CoppeliaSim pode ser executado em plataformas como Linux, Windows ou Mac. E a linguagem de programação pode ser C/C++, Python, Java, Lua, Matlab ou Octave [80].

Este simulador é utilizado para desenvolvimento rápido de algoritmos, simulações de automação de fábrica, monitoramento remoto, verificação dupla de segurança, entre outras aplicações [80].

5.1.3 Kelpie

O simulador Kelpie utiliza alguns conceitos do simulador WaveSim [16]. O Kelpie é capaz de contabilizar vários robôs virtuais aéreos, de superfície, subaquáticos e terrestres. De forma a assegurar a total integração com os sistemas de controlo de robôs autónomos, o Kelpie é totalmente compatível com o ROS e a sua arquitetura central é baseada no Gazebo [16]. Este simulador distingue-se do Gazebo pois fornece simulação física mais precisa e qualidade de renderização, utilizando as malhas geométricas dos modelos de veículos em vez de simples formas geométricas.

No Kelpie os robôs são simulados de acordo com as leis da física e renderizados em ambientes 3D. Os nós ROS deste sistema interagem uns com os outros através de mensagens e serviços de publicação e subscrição. Numa rede ROS, o Kelpie é apenas um nó com um conjunto reconfigurável de interfaces, ou seja, tópicos. Este nó interage com outros nós ROS, como atuadores e sensores de robôs virtuais, como se estivessem a interagir com os robôs reais [16].

A renderização gráfica em tempo real é gerida pelo OpenSceneGraph (OSG). Este é um conjunto de ferramentas de código aberto que é escrito em C++ e OpenGL. A simulação dos aspetos físicos do modelo é gerada pela biblioteca BulletPhysics. Este motor apresenta deteção de colisão discreta e contínua em 3D com diversas formas e também dinâmica corporal suave e rígida. A biblioteca osgBullet é utilizada para combinar os modelos baseados em OSG com o motor BulletPhysics. Para ser capaz de guardar e carregar um ficheiro *Simulation Description Format* (SDF) é utilizado um analisador *Extensible Markup Language* (XML) [16].

Na Figura 5.2 pode-se observar os principais componentes dos nós ROS responsáveis pelo Kelpie: (i) uma interface ROS, (ii) um renderizador de computação gráfica, (iii) um motor de física e (iv) um analisador XML.

5.2 Simuladores

Os simuladores virtuais permitem acompanhar o movimento do veleiro robótico, possibilitando a análise das suas acções ao longo de um determinado percurso. Além disso, permitem também fazer uma análise prévia do desempenho do veleiro

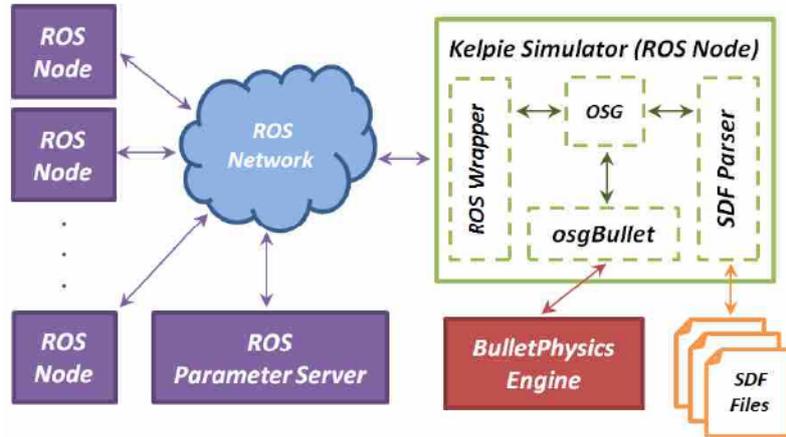


Figura 5.2: Arquitetura do sistema Kelpie [16].

robótico em determinadas circunstâncias, como vento fortes, marés inconstantes, podendo evitar possíveis danos.

Os principais requisitos a considerar num simulador para este projeto são [78]:

- um simulador modular de *Unmanned Surface Vehicle* (USV) gratuito que permita modelar diferentes barcos, cenários e estratégias;
- integração de um modelo de vento que afete as diversas partes do veleiro acima da linha d'água.

5.2.1 Morse

O MORSE é um simulador robótico académico baseado no Blender Game Engine e no mecanismo Bullet Physics. Este simulador é um projeto licenciado pela Berkeley Software Distribution (BSD) [81] e concentra-se em simulações 3D realistas de ambientes pequenos a grandes, tanto internos quanto externos, com várias plataformas robóticas simuladas simultaneamente.

O MORSE é compatível com o ROS, baseado na linguagem de programação Python e pode ser totalmente controlado a partir da linha de comandos. Além disso, fornece um grande conjunto de sensores padrão (câmaras, lidar, GPS, odometria, entre outros), atuadores (controladores de velocidade, controladores de *waypoints* de alto nível, controladores de juntas genéricas), bem como várias plataformas robóticas (veículo genérico de 4 rodas, PR2, entre outros) [82]. Este simulador não simula o vento ou as forças aerodinâmicas.

5.2.2 Gazebo

O Gazebo é um simulador dinâmico com a capacidade de simular com precisão e eficiência robôs em ambientes 3D complexos, com uma vasta interação dinâmica entre objetos [83].

O Gazebo apresenta as seguintes características [83]:

- simulação dinâmica - acesso a vários mecanismos de física de alto desempenho, incluindo Open Dynamics Engine (ODE), Bullet, Simbody e Dynamic Animation and Robotics Toolkit (DART);
- gráficos 3D avançados - usando o Object-Oriented Graphics Rendering Engine (OGRE), o Gazebo fornece renderização realista de ambientes, incluindo iluminação, sombras e texturas de alta qualidade;
- gere dados de sensores, opcionalmente com ruído, a partir de telémetro a laser, câmaras 2D / 3D, sensores de contacto, binário, entre outros;
- desenvolvimento de *plugins* personalizados para robôs, sensores e controlo. A estrutura modular do Gazebo permite a extensão dos seus principais recursos por meio de *plugins*. Estes fornecem acesso direto à API do Gazebo, podendo ser carregados na linha de comando ou especificados num arquivo *.sdf*.
- modelos de robô - são fornecidos muitos robôs, incluindo PR2, Pioneer2 DX, iRobot Create e TurtleBot;
- transporte TCP / IP - permite executar a simulação em servidores remotos estabelecendo interface com o Gazebo por meio de passagem de mensagens baseadas em *socket*;
- simulação na nuvem - recorre ao CloudSim de forma a executar o Gazebo no Amazon AWS e GzWeb com o intuito de interagir com a simulação por meio de um navegador;
- ferramentas de linha de comando - ferramentas abrangentes de linha de comando facilitam a introspecção e controlo da simulação.

5.2.3 USVSim

O simulador USVSim modela e simula as forças que atuam em modelos USV de vela tradicional em diferentes cenários [17].

Um veleiro precisa resistir a forças ambientais previsíveis impulsionadas por ondas, vento e correntes de água. E, por isso, é indispensável um sistema de controlo robusto. Além disso, é necessário um ambiente de simulação com modelos

realistas de veleiros que possam ser testados recorrendo a estratégias de controlo definidas, sob diferentes graus de perturbações ambientais.

A arquitetura principal do sistema USVSim é constituída pelo UWSIM e Gazebo. Através de alguns *plugins* é possível simular de forma realista os efeitos das ondas, vento e correntes de água em diversos tipos de barcos. Na Figura 5.3 pode-se observar a arquitetura do sistema e a interação entre o *plugin* do Gazebo denominado *usv_sailing_plugin* e os nós ROS *wind_current* e *water_current*.

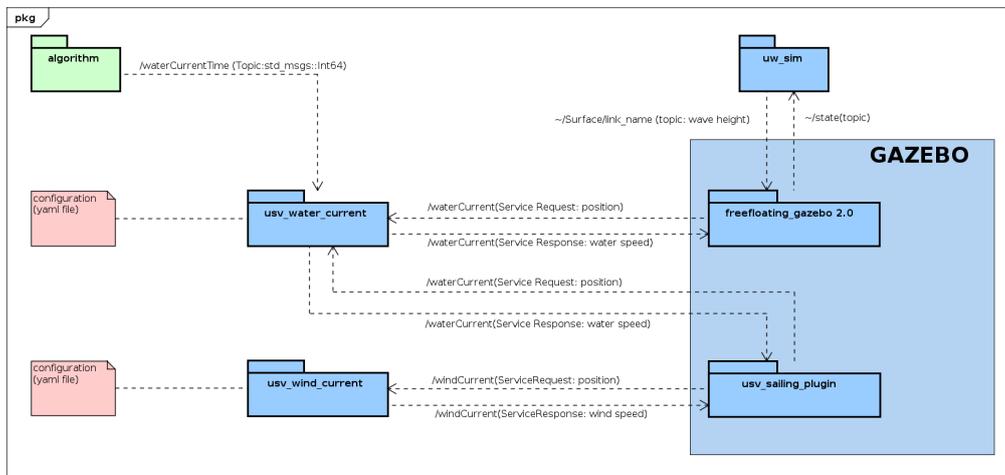


Figura 5.3: Arquitetura do Sistema USVSim [17].

O simulador USVSim dispõem de quatro tipos de barcos pré-configurados diferentes, ilustrados na Figura 5.4, tais como: (i) barco a leme com uma propulsor e um leme, sendo das configuração mais comuns em barcos, (ii) aerobarco composta por um propulsor acima do casco, este modelo apresenta vantagens quando navegado em águas rasas, (iii) barco diferencial com dois propulsores sob a superfície da água, este modelo possui a manobrabilidade mais simples e (iv) veleiro com uma vela, um leme e uma quilha.

Além disso, o casco de todos os modelos apresentados acima foram subdivididos em 6 partes, conforme demonstrado na Figura 5.5. As ondas afetam a flutuabilidade do modelo de tal forma que sentiu-se a necessidade da subdivisão do casco de forma ao barcos apresentarem movimentos mais realistas. No caso, de se pretender mais realismo, pode-se subdividir o casco em mais parte, recorrendo a ferramentas geométricas, tais como o Blender para modelar cada parte do casco.

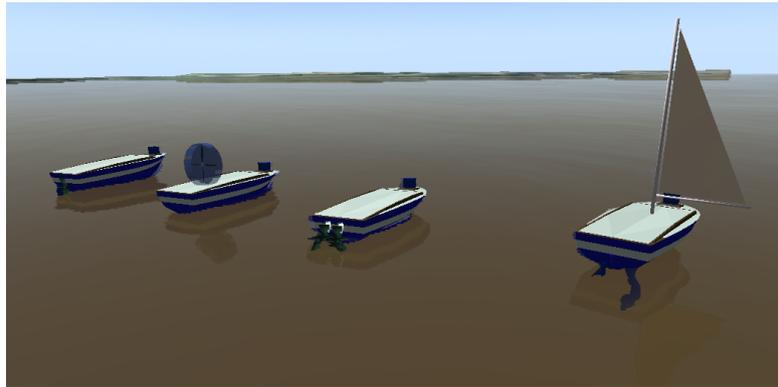


Figura 5.4: Barco a leme, aerobarco, barco diferencial e veleiro do simulador USVSim [17].

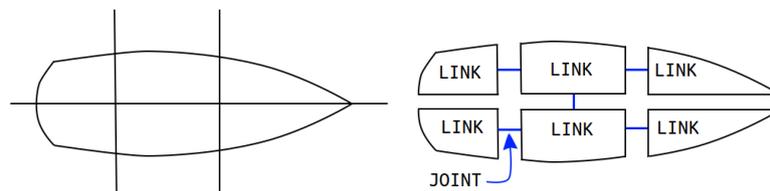


Figura 5.5: Subdivisão do casco em 6 partes [17].

5.3 Análise das Ferramentas de Simulação

Nesta secção são apresentadas as principais características das ferramentas de simulação estudadas anteriormente, resumidas na Tabela 5.1.

Os simuladores 3D apresentados são todos compatíveis com o ROS. No entanto, os simuladores mais comumente utilizados para simular veículos robóticos de superfície são o Gazebo e o Morse. Segundo Nathan *et. al* [84], com base em testes com ambas as ferramentas para as mesmas simulações, concluiu-se que o Morse tem um desempenho ligeiramente superior ao Gazebo. A vantagem mais relevante do Gazebo é a integração total na estrutura ROS. O Morse também suporta ROS, mas em comparação com o Gazebo, o suporte é menos desenvolvido. A base de simulação será calculada e programada no ROS e o Gazebo apresenta o motor de física mais extenso do que o Morse, possibilitando um maior grau de realismo para a simulação. Enquanto o Morse recorre ao Bullet para modelação, o Gazebo utiliza ODE e Dart como dois motores adicionais.

Tabela 5.1: Características das diversas ferramentas de simulação.

	UWSim	CoppeliaSim	Kelpie	Morse	Gazebo	USVSim
Linguagem	C++	Python Java C/C++	C++	Python C++ Java	C++	C++
Sistema Operativo	Linux	Linux Mac OS Windows	Linux	Linux Mac OS	Linux Mac OS	Linux Mac OS
Motor de Física	Bullet	ODE Bullet Vortex Newton	Bullet	Bullet	ODE Bullet Dart Simbody	ODE Bullet Dart Simbody
Motor de Renderização	OSG	Internal External	OSG	Blender	OGRE	OGRE OSG
Integração ROS	Sim	Sim	Sim	Sim	Sim	Sim
Formato Suportado	URDF	URDF	SDF	URDF	URDF SDF	URDF

5.4 Conclusão do Capítulo

Neste capítulo estudaram-se alguns ambientes de simulação e simuladores compatíveis com o ROS, destacando as suas particularidades e limitações. Optou-se pela utilização do USVSim, que inclui bibliotecas do simulador UWSim, desenvolvido sobre o ambiente de simulação Gazebo que permite simular de forma realista os efeitos das ondas em diversos tipos de barcos.

Capítulo 6

Ambiente de Desenvolvimento

Este capítulo apresenta, após análise detalhada, o ambiente de desenvolvimento deste projeto, a integração com o ROS, os plugins reutilizados e desenvolvidos.

6.1 Introdução

Os ambientes de desenvolvimento são *softwares* computacionais que reúnem características e ferramentas de apoio ao desenvolvimento de *software*. Diversos destes ambientes incluem, normalmente: (i) editores, que permitem editar o código-fonte nas linguagens suportadas pelo ambiente de desenvolvimento, tais como C++, Python, entre outros, (ii) compiladores, que permitem compilar o código-fonte do programa, (iii) *debuggers*, que consistem em depurar os defeitos no código-fonte, entre outras ferramentas.

A plataforma robótica de vela rígida tem como principal objetivo ser robusta, dado que ao longo da sua navegação estará sujeita a condições atmosféricas adversas, tais como ventos fortes e marés inconstantes, viável e simples de controlar. Em função disso, sentiu-se a necessidade de desenvolver um sistema de controlo vigoroso e confiável.

Na secção seguinte vai-se abordar o sistema ROS responsável pelo *software* implementado neste projeto, o seu princípio de funcionamento, assim como os conceitos mais utilizados.

6.2 ROS

O ROS é uma coleção de ferramentas e bibliotecas que visam simplificar a tarefa de conceber comportamentos robustos e complexos de robôs, numa ampla

diversidade de plataformas robóticas, devido à sua flexibilidade, robustez e modularidade.

O ROS é um *middleware* de *open-source* utilizado para o desenvolvimento de sistemas robóticos. Este permite aos utilizadores estabelecerem um ambiente onde é possível colaborar no desenvolvimento do *software* robótico a nível global, permitindo a troca de ideias entre investigadores. A preferência do ROS para o desenvolvimento de *software* na área da robótica tem vindo a crescer exponencialmente, dado que este apresenta diversas vantagens, tais como [85]:

- processo distribuído: é programado na forma de unidades mínimas de processos executáveis, denominados nós, e cada processo é executado de forma independente. Além disso, os nós conseguem comunicar entre si tendo por base um nó *master*, designado de *roscore*;
- gestão de pacotes: vários processos com a mesma finalidade são geridos como um pacote para que seja fácil de usar e desenvolver, bem como compartilhar, modificar e redistribuir;
- independência do *hardware*: permitindo o desenvolvimento de *software* de maior complexidade sem a necessidade de existir algo implementado fisicamente;
- repositório público: cada pacote pode ser tornado público, com o intuito de compartilhar com outras pessoas, por exemplo no GitHub;
- suporte a várias linguagens de programação: o ROS fornece uma biblioteca para oferecer suporte a várias linguagens de programação, tais como C++ e Python.

A nível do controlo, um dos principais objetivos consiste em estabelecer comunicação entre o ambiente de simulação Gazebo e o ROS, de forma a que seja possível analisar o comportamento do veleiro a partir de informações provenientes dos nós ROS.

6.2.1 Comunicação

A comunicação em ROS baseia-se em serviços, ou na criação de ligações entre *publishers/subscribers* através de um tipo específico de mensagem. O ROS permite que nós em paralelo possam comunicar entre si, suportando a troca de mensagens sob a forma de serviços ou tópicos. Cada nó ROS pode informar ao *master* o tópico que pretende subscrever ou publicar, e caso exista correspondência a uma outra publicação/subscrição, é então criada uma ligação direta entre o *publisher* e o *subscriber*.

A Figura 6.1 [18] representa o funcionamento do mecanismo dos tópicos do ROS. Como dito anteriormente, quando um *publisher* (a) e um *subscriber* (b) estão registados no mesmo tópico “ttype”, o *subscriber* recebe o endereço de todos os *publishers* (c). O *subscriber* pode estabelecer uma ligação direta com cada um dos *publishers*, sendo que estes passarão a enviar a informação pretendida diretamente para o *subscriber* (d). No entanto, vários nós podem publicar e subscrever ao mesmo tópico, como consta no esquema (e) [18].

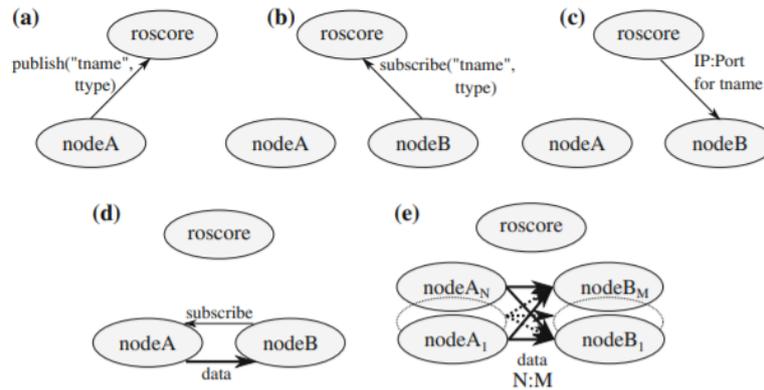


Figura 6.1: Funcionamento do mecanismo de tópicos ROS [18].

6.2.2 Conceitos e Terminologias

No ROS existem diversos conceitos importantes, tais como:

- **master**: permite a conexão entre os nós e a comunicação de mensagens, como tópicos e serviços, através da variável `ROS_MASTER_URI` também referenciado como *roscore*. Sem o *master*, os nós não se conseguem localizar, impossibilitando a comunicação;
- **nó**: refere-se à menor unidade de processamento em execução no ROS. Este último recomenda a criação de um único nó para cada finalidade, de forma a facilitar a sua reutilização;
- **tópico**: canal de comunicação assíncrono e unidireccional;
- **mensagens**: estruturas de dados específicos, tais como números inteiros, *floating point*, booleano, etc., transmitidos entre nós. As mensagens também suportam os *arrays*;
- **serviço**: comunicação bidireccional síncrona entre o cliente e o servidor. O cliente solicita um serviço referente a uma tarefa específica e o servidor é responsável por responder às solicitações;

- **bags**: é um formato de armazenamento e reprodução de dados de mensagens (tópicos). A sua extensão é `.bag`;
- **roslaunch**: permite a iniciação de vários nós ROS, a partir da linha de comandos. Além disso, permite a execução de nós com funções adicionais, como alteração de parâmetros ou nomes de nós.

As ferramentas ROS facilitam a compreensão, o monitoramento e a depuração de problemas à medida que ocorrem. Essas ferramentas oferecem suporte à introspecção, depuração e visualização do estado do sistema. Todas as ferramentas são acessíveis por linha de comando, com funcionalidades semelhantes ou estendidas e também estão incluídas interfaces gráficas para melhor interpretação, como *rviz* e *rqt*. O *rqt* é um *framework* baseado em Qt com *plugins rqt* já incluídos como: `rqt_graph`, `rqt_plot`, `rqt_topic`, `rqt_publisher`, `rqt_reconfigure`, `rqt_console` e `rqt_bag`. Os *plugins rqt* também podem ser estendidos e criados com novas funcionalidades.

O ROS fornece bibliotecas e ferramentas específicas para robôs. O seu espaço de trabalho é descrito pela seguinte arquitetura:

- **packages**: o *software* no ROS é organizado em pacotes. Os pacotes permitem organizar o *software*. Um pacote pode conter nós ROS, uma biblioteca independente do ROS, um conjunto de dados, arquivos de configuração, entre outros. Os pacotes são acessíveis de criar manualmente ou com ferramentas como `catkin_create_pkg`. Um pacote ROS é meramente um diretório descendente de `ROS_PACKAGE_PATH` que contém um arquivo `package.xml`;
- **metapackages**: as *metapackages* são pacotes especializados em ROS (e `catkin`). Elas não instalam arquivos (exceto o seu manifesto `package.xml`) e não contêm nenhum teste, código, arquivo ou outros itens normalmente encontrados em pacotes;
- **package manifests**: o *package manifest* é um arquivo XML denominado `package.xml` que deve ser incluído em qualquer pasta raiz de pacote compatível com `catkin`. Este arquivo define propriedades sobre o pacote, como o nome do pacote, autores e dependências de outros pacotes `catkin`. As dependências do pacote do seu sistema são declaradas em `package.xml`;
- **message types**: as descrições das mensagens, armazenadas na pasta específica das mensagens do pacote, definem as estruturas de dados das mensagens enviadas no ROS;

- **service types**: as descrições de serviço, armazenadas na pasta específica dos serviços de pacote, definem as estruturas de dados de solicitação e resposta para serviços em ROS.

6.3 Gazebo

O ambiente de simulação Gazebo utiliza uma arquitetura distribuída com bibliotecas separadas para simulação física, renderização, interface de utilizador e comunicação. A simulação é separada por dois programas executáveis [86]:

- **gzserver**: simula física, renderização e sensores. Ele analisa um arquivo de descrição do mundo e, em seguida, simula o mundo recorrendo ao mecanismo de física e sensor;
- **gzclient**: fornece uma GUI para visualização e interação/modificação da simulação.

A gestão de tópicos e a pesquisa de nomes são realizados pelo Gazebo Master. Um único mestre pode lidar com várias simulações de física, geradores de sensores e GUI.

A biblioteca de física fornece componentes de simulação fundamentais, incluindo corpos rígidos, formas de colisão e restrições de articulação de juntas. As bibliotecas de física integradas de *open-source* são: ODE, Bullet, Simbody, DART, com a possibilidade de carregar modelos de robôs com qualquer um desses motores.

A biblioteca de renderização utiliza OGRE, que fornece uma interface simples para renderizar cenas 3D para as bibliotecas de GUI e sensores. A biblioteca GUI usa Qt para criar *widgets* gráficos para simulação.

Todas as bibliotecas no simulador Gazebo suportam *plugins*. Esses *plugins* têm acesso a bibliotecas e podem adicionar novos recursos sem usar o sistema de comunicação.

6.4 Integração Gazebo-ROS

A integração entre o Gazebo e o ROS é realizada com a *metapackage* `gazebo_ros_pkgs` que inclui [87]:

- **gazebo_ros**: envolve `gzserver` e `gzclient` recorrendo a dois *plugins* Gazebo que fornecem a interface ROS necessária para mensagens, serviços e reconfiguração dinâmica;

- **gazebo_msgs:** mensagens e estruturas de dados de serviço para interagir com o ROS;
- **gazebo_plugins:** *plugins* do Gazebo independentes do robô, sensores e motores (juntas, força, modelo).

A Figura 6.2 representa a arquitetura da integração Gazebo-ROS. O *plugin* `gazebo_ros_api_plugin`, localizado dentro do pacote `gazebo_ros`, inicializa um nó ROS chamado “gazebo”. Este pacote habilita a interface entre o ROS e o Gazebo, permitindo também a criação de nós ROS dentro dos *plugins* do Gazebo.

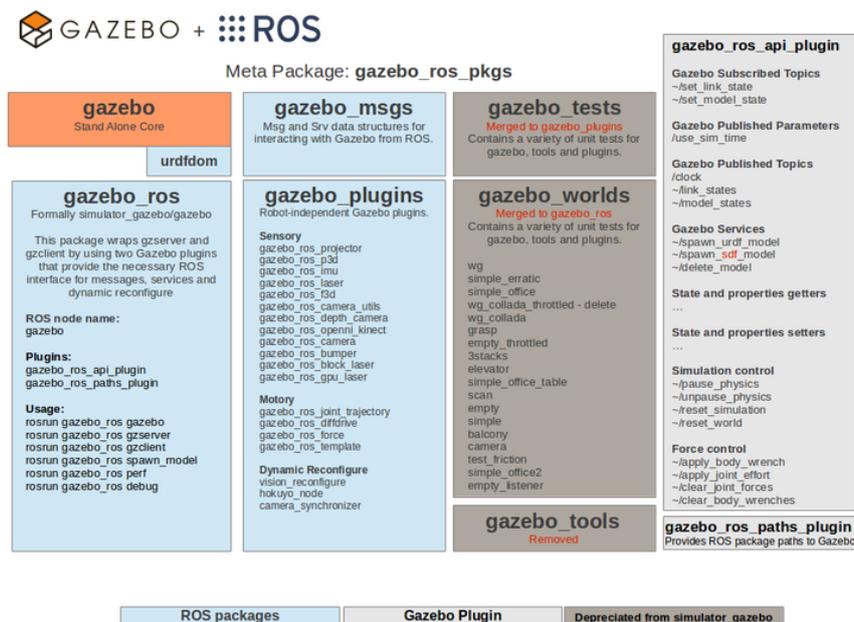


Figura 6.2: Arquitetura da integração Gazebo-ROS [19].

Um segundo *plugin*, `gazebo_ros_paths_plugin`, também disponível no pacote `gazebo_ros`, permite que o Gazebo encontre recursos ROS.

6.5 Plugins

Um *plugin* no Gazebo é um código que é compilado como uma biblioteca compartilhada e inserido diretamente na simulação. O *plugin* tem acesso direto a todas as funcionalidades do Gazebo através das classes C++/Python padrão [88].

Os *plugins* são úteis porque [88]:

- permitem que os desenvolvedores controlem quase qualquer aspecto do Gazebo;
- são rotinas independentes, sendo facilmente compartilhadas;
- podem ser inseridos e removidos de um sistema em execução.

As versões anteriores do Gazebo utilizavam controladores. No entanto, os *plugins* são mais flexíveis e permitem que os utilizadores decidam que funcionalidades incluir nas suas simulações. Os *plugins* são usados quando [88] (i): deseja-se alterar uma simulação, como responder a eventos ou inserir novos modelos dado um conjunto de pré-condições, (ii) pretende-se uma interface rápida para o Gazebo sem a sobrecarga da camada de transporte, entre outros.

Todos os *plugins* devem estar dentro do *namespace* do Gazebo. Relativamente aos tipos de *plugins* existentes atualmente são: (i) World, (ii) Model, (iii) Sensor, (iv) System, (v) Visual e (vi) GUI. Uma função obrigatória é `Load`, que recebe o arquivo SDF com todas as descrições da plataforma robótica.

No final, o *plugin* é registado dentro do simulador por meio de uma macro. Na inicialização, o Gazebo analisa o arquivo SDF, localiza o *plugin* e carrega o código. Cada *plugin* é anexado a diferentes partes da descrição do ambiente, dependendo do tipo de *plugin*.

Os *plugins* do Gazebo oferecem aos seus modelos *Unified Robot Description Format* (URDF) maior funcionalidade, podendo vincular mensagens ROS e chamadas de serviço [89] assim como adicionar novas funcionalidades à simulação.

Os *plugins* do Gazebo usados para a simulação foram o *plugin* de forças aerodinâmicas `LiftDragPlugin`, o *plugin* que simula a flutuabilidade `freefloating_gazebo_fluid`, o que lida com o controlo de propulsores e juntas `freefloating_gazebo_control`, o `hector_gazebo_plugins` para os sensores [90] e outros *plugins* especificamente criados para este projeto. Os *plugins* desenvolvidos para esta dissertação foram: (i) o `usv_flap` que faz a interface do ROS com as juntas da cauda e do leme, (ii) o *plugin* `anemometer` que simula o sensor de vento recorrendo à velocidade e direção do veículo e do vento real e (iii) o *plugin* `usv_control` que estabelece o controlo da cauda e do leme.

6.5.1 LiftDragPlugin

O Gazebo simula as forças aplicadas num objeto imerso num fluido e aplica-as diretamente nos elos do objeto. Em particular, os fenómenos de sustentação (*lift*) e resistência aerodinâmica (*drag*) são fundamentais para veículos subaquáticos, aquáticos e aéreos [21].

O `LiftDragPlugin` simula as forças aerodinâmicas de um perfil alar (o binário gerado pelo vento no perfil alar), pois foi desenvolvido para aviões. Dado que o

perfil alar num veleiro está posicionado verticalmente, o *plugin* irá calcular o momento de guinada. Em suma, o `LiftDragPlugin`, além de gerar forças de sustentação e arrasto, também alinha a vela asa na direção do vento. No entanto, este *plugin* foi criado para simular essas forças nos aviões ou *drones* de asa fixa em condições sem vento. Noutras palavras, este *plugin* assume que a velocidade do vento no mundo corresponde à velocidade linear do veículo, que, neste caso, está apenas parcialmente correto. O *plugin* foi modificado para adicionar a velocidade do vento no mundo à velocidade linear do veículo. Por exemplo, se o veículo está a mover-se para norte com uma velocidade linear de 1 m/s e o vento sopra de norte a 3 m/s, então as duas velocidades lineares são adicionadas, gerando uma velocidade de vento aparente de 4 m/s. Sem essa mudança, a velocidade aparente do vento permaneceria 1 m/s. Por fim, o *plugin* foi modificado de forma a estabelecer comunicação com o ROS e adaptado para subscrever um tópico ROS que transmite a velocidade real do vento nos eixos x e y . Na Figura 6.3 pode-se observar um diagrama que detalha o processamento do *plugin* modificado. Para usar este *plugin*, as características da vela asa devem ser descritas usando o formato mostrado na Figura 6.4, onde as curvas do perfil alar são aproximadas em linhas e as inclinações obtidas são usadas como parâmetros na importação do *plugin*.

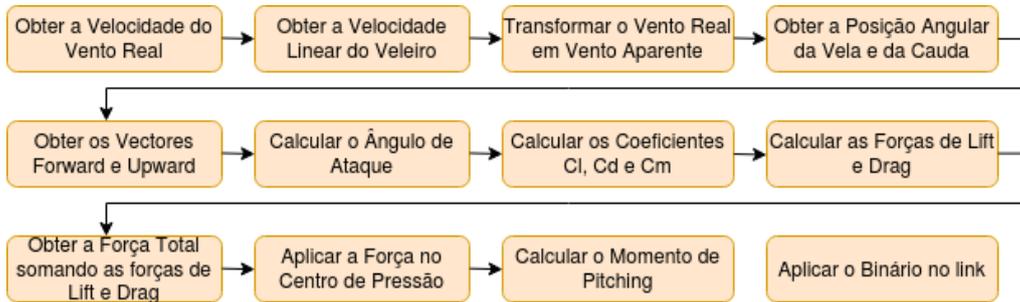


Figura 6.3: Diagrama de blocos `LiftDragPlugin` [20].

Com o intuito dos parâmetros para o `LiftDragPlugin` serem definidos corretamente, foi seguido um tutorial Gazebo para obter as seguintes constantes [21]:

- *link_name*: nome do *link* afetado pelo grupo de propriedades de sustentação/arrasto;
- *fluid_density*: densidade do fluido em que o modelo está imerso;
- *area*: área de superfície da vela;
- *a0*: α inicial ou ângulo de ataque inicial.

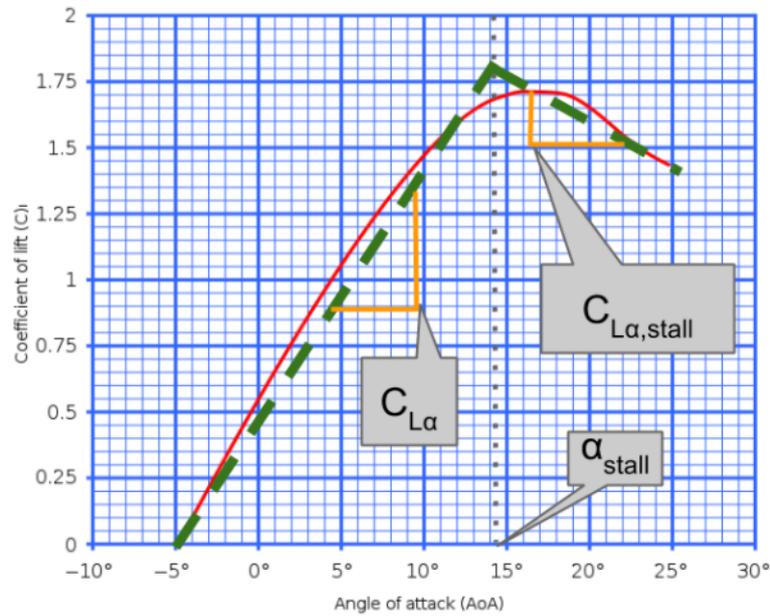


Figura 6.4: Relação entre o ângulo de ataque (α) e o coeficiente de sustentação (C_L) [21].

- c_{la} : inclinação da primeira parte da curva do coeficiente de sustentação *vs* α ;
- c_{da} : a proporção do coeficiente de arrasto e inclinação α antes do ângulo de ataque crítico;
- cp : centro de pressão;
- *forward*: vector que representa a direção do movimento;
- *upward*: vector que representa a direção de sustentação.

Os *plugins* podem ser carregados no modelo ou mundo. Um exemplo de como carregar um *plugin*, neste caso o `LiftDragPlugin`, é apresentado na Listagem 6.1. Os parâmetros obtidos a partir das características do perfil alar EPPLER 479 foram convertidos de acordo com o exemplo apresentado na Figura 6.4.

De forma a definir a vela principal na posição desejada, em primeiro lugar, a orientação da cauda deve ser controlada corretamente e ser capaz de fornecer estabilidade à vela asa. Com base nas dimensões escolhidas da cauda, apresentadas no Capítulo 4, o ângulo de ataque da vela asa principal pode ser definido em função do ângulo da cauda. Existem três equações para calcular o ângulo da

Listagem 6.1: Exemplo de *plugin* SDF.

```

1 <plugin name="usv_sail_plugin" filename="libfoil_dynamics_plugin.so">
2   <link_type>sail</link_type>
3   <a0>0</a0>
4   <cla>5.9917</cla>
5   <cda>0.5526</cda>
6   <cma>0.0</cma>
7   <alpha_stall>0.2181</alpha_stall>
8   <cla_stall>-3.4149</cla_stall>
9   <cda_stall>0.6424</cda_stall>
10  <cma_stall>0</cma_stall>
11  <cp>0.375 0 1.25</cp>
12  <joint_name>sail_joint</joint_name>
13  <area>1.88</area>
14  <fluid_density>1.2041</fluid_density>
15  <forward>-1 0 0</forward>
16  <upward>0 1 0</upward>
17  <link_name>sail</link_name>
18  <fluidVelocity>${windType}</fluidVelocity>
19 </plugin>

```

cauda de forma posicionar a vela principal na posição pretendida. Primeiro, a razão de comprimento adimensional d_r e a razão da área adimensional A_r devem ser calculadas de acordo com a Equação 6.1 e a Equação 6.2 [91]. Nessas equações, d é a distância do quarto da corda da vela principal ao quarto da corda da cauda, C_m o comprimento da corda da vela principal, A_t a área da cauda e A_m a área da vela principal. Com as duas relações, é possível calcular o coeficiente do momento, C_M , em torno do ponto de pivô, conforme demonstrado na Equação 6.3 [91], onde $C_{L\alpha}$ é o coeficiente de sustentação correspondente ao ângulo de ataque da cauda, AR_m a proporção da vela principal (envergadura ao quadrado a dividir pela área da vela), α o ângulo de ataque da vela principal e γ_{Tail} o ângulo da cauda. Na Tabela 6.1 podem-se observar os valores dos parâmetros.

$$d_r = \frac{d}{C_m} \quad (6.1)$$

$$A_r = \frac{A_t}{A_m} \quad (6.2)$$

O conjunto vela asa-cauda deve ter a massa equilibrada em torno do mastro para que os ângulos de inclinação e rotação não induzam mudanças no ângulo de ataque e afetem o estado de equilíbrio do sistema [28]. Um conjunto pesado de vela asa-cauda resultaria num aumento do ângulo de ataque com o ângulo

$$C_M = -d_r * A_r * (C_{L\alpha}) \left[\frac{AR_m}{AR_m + 2} * \alpha + \gamma_{Tail} \right] \quad (6.3)$$

Tabela 6.1: Dimensionamento de parâmetros.

d (m)	C_m (m)	A_t (m ²)	A_m (m ²)	AR_m	α (°)
1,75	0,75	0,23	1,88	3,32	8,5

roll, induzindo assim a instabilidade em condições de bolina. Uma plataforma desequilibrada pode induzir movimento de rotação para o veleiro. Além disso, o centro aerodinâmico de pressão deve ser deslocado ligeiramente do eixo z . Se o momento de inércia I_z for baixo, reduz as cargas e facilita a rotação da asa (I_z descreve a inércia de uma massa a girar em torno de um eixo). Portanto, quanto menor a massa da vela asa e a distância ao eixo, mais fácil será a sua rotação [92].

6.5.2 FreeFloating

O `FreeFloating` é uma ferramenta de simulação direcionada para robôs subaquáticos. Este *plugin* pode ser utilizado pelo simulador UWSim e Gazebo.

O pacote `FreeFloating` cria dois *plugins* no Gazebo:

- `freefloating_gazebo_fluid`, *plugin* mundo, que simula a flutuabilidade e a superfície da água.
- `freefloating_gazebo_control`, *plugin* modelo, que lida com o controle de baixo nível através do esforço do propulsor e juntas, a fim de controlar as plataformas robóticas.

6.5.3 hector_gazebo_plugins

O *plugin* `hector_gazebo_plugin` simula vários sensores, como recetores GNSS e IMU. Este último *plugin* não exigiu nenhuma modificação, pois funciona corretamente e publica os dados dos sensores nos tópicos ROS.

6.5.3.1 GNSS

O *plugin* GNSS simula um recetor GNSS que está conectado ao veleiro. Este *plugin* publica mensagens do tipo `sensor_msgs/NavSatFix` com a posição e altitude do veleiro em coordenadas *World Geodetic System 84* (WGS84). O ponto de referência que corresponde à origem do *frame* do Gazebo pode ser configurado

usando os parâmetros XML. A conversão entre as coordenadas do Gazebo e WGS84 é realizada recorrendo à projeção equirretangular simples, que é precisa o suficiente, no caso de não se afastar do ponto de referência configurado e não se pretender recorrer ao *plugin* para regiões polares.

Este *plugin* publica no tópico:

- `/sailboat/fix`: mensagens do tipo `sensor_msgs/NavSataFix`, a posição GNSS simulada em coordenadas WGS84 (latitude, longitude e altitude).
- `/sailboat/fix_velocity`: mensagens do tipo `geometry_msgs/Vector3Stamped`, o vector de velocidade GNSS em coordenadas *North-West-Up* (NWU).

6.5.3.2 IMU

O *plugin* IMU simula uma unidade de medição inercial (IMU) afetada por ruído gaussiano e desvio aleatório de baixa frequência. A orientação retornada emite um simples *Attitude and Heading Reference System* (AHRS) recorrendo a taxas e acelerações.

Este *plugin* publica no tópico:

- `/sailboat/imu`: mensagens do tipo `sensor_msgs/Imu`, as medições IMU simuladas e a orientação estimada em relação ao *frame* do Mundo.

6.5.4 usv_flap

O *plugin* `usv_flap` é composto por um nó ROS que subscreve os tópicos do ângulo da cauda e o ângulo de direção do leme. Quando o *plugin* recebe um retorno de chamada desses tópicos, ele define a posição da junta para a posição angular transmitida.

6.5.5 anemometer

O *plugin* `anemometer` simula o sensor de vento, que está conetado ao topo do mastro. Este *plugin* subscreve o tópico do vento e calcula a velocidade e direção do vento aparente, tendo por base as informações da direção e velocidade do vento real e da velocidade e orientação do veleiro no mundo. Por fim, utiliza um nó ROS para publicar mensagens do tipo `geometry_msgs/Vector3` com a velocidade e a direção do vento aparente.

6.5.6 usv_control

O *plugin* `usv_control` subscreve os nós ROS relativos à velocidade e direção do vento aparente e à orientação da vela asa, de forma a calcular o ângulo de

ataque da vela asa. Além disso, subscreeve também o nó ROS com informações da posição e velocidade linear do veleiro. Este *plugin* é responsável pelo controlador PD, para o controlo da cauda, e pelo controlador P, para controlo da direção do veleiro.

6.6 Conclusão do Capítulo

Este capítulo apresenta o ambiente de desenvolvimento do projeto, assim como a integração do Gazebo com o ROS. Os *plugins* `LiftDragPlugin`, `FreeFloating` e `hector_gazebo_plugins` foram reutilizados e, estes dois últimos não sofreram quaisquer alterações. Os *plugins* desenvolvidos foram o `usv_flap`, `anemometer` e o `usv_control`.

Capítulo 7

Desenvolvimento da Simulação

Neste capítulo é apresentado o modelo do veleiro 3D, a descrição do ambiente e o sistema de controlo deste projeto.

7.1 Simulação da Plataforma

A simulação foi desenvolvida com o simulador Gazebo e o *middleware* ROS. O modelo da vela rígida e da cauda foram desenvolvidos no Autodesk Fusion 360 e as *meshes* criadas foram integradas no Gazebo, recorrendo aos ficheiros SDF e XML. Os modelos SDF descrevem: (i) os *links*, correspondendo a partes do corpo, incluindo elementos de colisão (geometria), visual (visualização) e inércia (dinâmica), (ii) as juntas definidas entre dois *links*, podendo ser de vários tipos, tais como: rotacionais, prismáticas, entre outras, (iii) os sensores que recolhem dados para os *plugins*; e (iv) os *plugins* que controlam o comportamento do modelo.

7.2 Modelo 3D do Veleiro

As partes constituintes do veleiro são modeladas tendo por base três ficheiros diferentes, (i) o ficheiro Collada que fornece todas as informações sobre a colisão, (ii) o ficheiro Wavefront OBJ responsável pela visualização dos *links* e (iii) o ficheiro *Material Template Library* (MTL) que contém diversos parâmetros permitindo definir as texturas dos objetos.

O veleiro é constituído, como representado na Figura 7.1, pelo casco (importado do pacote USVSim) denominado `base_link`, que se conecta a outros *links*, tais como: `back_l_link`, `back_r_link`, `center_l_link`, `center_r_link`, `front_l_link` e `front_r_link` que formam juntos o casco, sendo dividido em 6

partes de forma a tornar o efeito de fluatibilidade das ondas o mais realista possível. O `hokuyo_link` que é uma caixa fixa na parte frontal superior do veleiro, a quilha representada pelo `link keel` e o leme pelo `rudder` são interligados na parte inferior do casco. O modelo da vela representado pelo `link sail` é conectado ao casco do veleiro e o modelo da cauda, representado pelo `link tail`, é conectada à vela. Ambos os modelos, vela e cauda, foram criados no AutoDesk Fusion 360 *Computer-Aided Design* (CAD) e encontram-se ilustrados na Figura 7.2 e na Figura 7.3, respetivamente.

Além dos modelos da vela asa e da cauda, criados no Autodesk Fusion 360, foi desenvolvido o modelo do sensor anemómetro, com o objetivo de indicar a direção do vento aparente, assim como influenciar a simulação. Com base no estudo desenvolvido no Capítulo 3 concluiu-se que, para a elaboração deste projeto, um sensor de vento constituído por um sensor de velocidade do vento e um sensor de direção do vento é o ideal. E, por isso, foi desenvolvido um sensor de vento com ambas as funcionalidades, conforme demonstrado na Figura 7.4. Na Figura 7.5 pode-se observar o sensor anemómetro implementado no veleiro.

Os *links* e as juntas do veleiro são definidos nos ficheiros URDF e *XML Macros* (Xacro) do veleiro:

- " `/catkin_ws/install_isolated/share/usv_sim/urdf`" é o caminho do ficheiro URDF.
- " `/catkin_ws/src/usv_sim_lsa/usv_sim/xacro`" é o caminho do ficheiro Xacro.
- " `/catkin_ws/src/usv_sim_lsa/usv_sim/meshes/simpleHull13`" é o caminho que contém os ficheiros Collada, Wavefront OBJ e MTL de cada parte do veleiro.

O modelo 3D do veleiro foi criado no *software* AutoDesk Fusion 360 e importado para o Gazebo recorrendo aos ficheiros SDF e XML, correspondentes. O modelo do veleiro é composto por vários componentes (*links*) e conexões (juntas). Os ficheiros SDF, onde todos os componentes do veleiro e propriedades das juntas são importados é chamado de ficheiro do modelo. Na Listagem 7.1 é apresentado um exemplo de importação de uma *mesh* 3D para o Gazebo, que contém um conjunto de parâmetros: (i) o nome do *link*, (ii) a posição do *link* em relação ao *link* da base (neste caso, o *link* da base é o casco), (iii) as propriedades inerciais (como a massa e a matriz inercial), (iv) as propriedades de colisão do *link* (neste caso, a zona de colisão é a mesma que o formato do modelo) e (v) o formato visual do *link* [93].

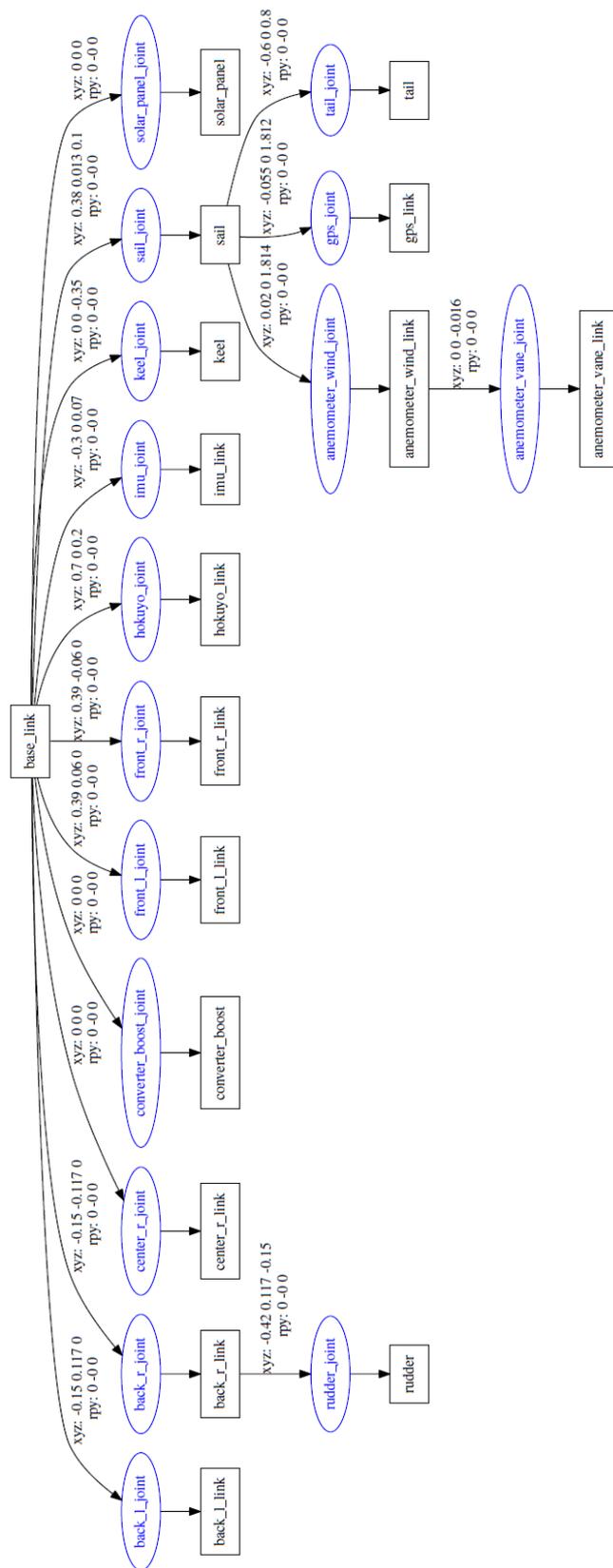


Figura 7.1: Modelo do veleiro no formato URDF.

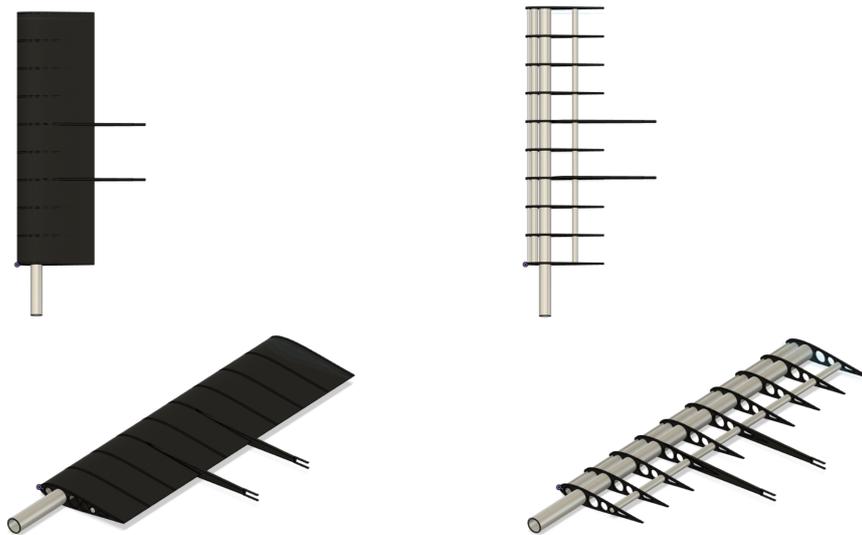


Figura 7.2: Modelo da vela criado no *software* Fusion 360.

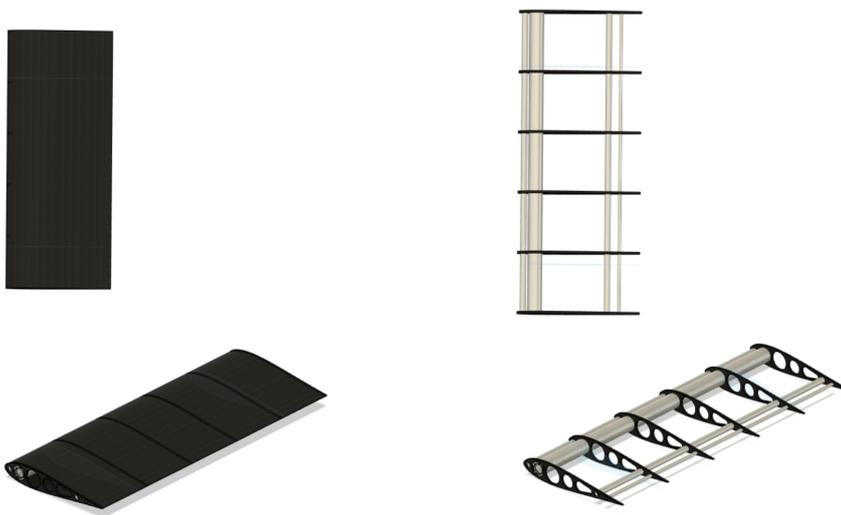


Figura 7.3: Modelo da cauda criado no *software* Fusion 360.

Na Listagem 7.2 pode-se observar como se especifica uma junta entre dois *links*, incluindo o nome da junta, o tipo de junta (uma vez que se pretende que a vela gire, o tipo de junta é rotativa), o nome dos *links* pai e filho (neste caso, o casco e a vela, respetivamente), o valor de amortecimento físico da junta, a posição da junta, o eixo de rotação (neste caso, a junta gira no eixo z) e a especificação do máximo esforço das juntas, a velocidade e os seus limites [94].

Por fim, o ficheiro SDF do mundo carrega o modelo do veleiro, bem como as características ambientais, como o sol, e os *plugins* relacionados com o mundo.

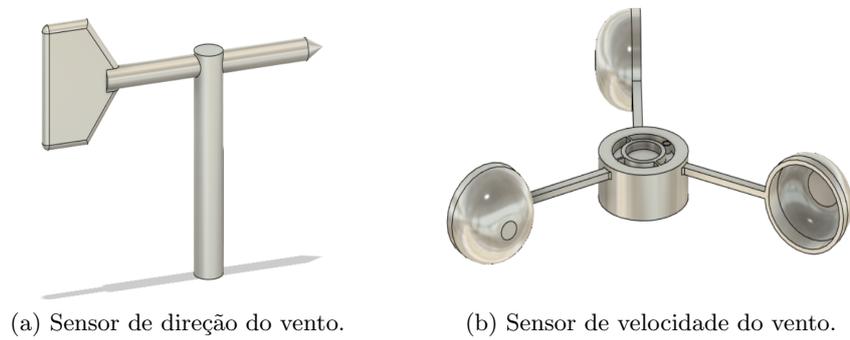


Figura 7.4: Sensor anemómetro.



Figura 7.5: Sensor anemómetro implementado no veleiro.

Na Figura 7.6 pode-se observar o modelo do veleiro no ambiente de simulação Gazebo, constituído por 17 *links* e 16 juntas.

7.3 Definição do Ambiente de Simulação

7.3.1 Linguagens

O ambiente de simulação define-se através de um conjunto de ficheiros em SDF, URDF e Xacro.

Listagem 7.1: Exemplo de código do modelo SDF

```

1 <link name="sail">
2   <inertial>
3     <origin rpy="0 0 0" xyz="0 0 0"/>
4     <mass value="{37.659*scale_model}"/>
5     <inertia
6       ixx="{23.671*scale_model}"
7       ixy="{-0.162*scale_model}"
8       iyy="{1.561*scale_model}"
9       ixz="{-0.0000434*scale_model}"
10      iyz="{-0.0000164*scale_model}"
11      izz="{25.146*scale_model}"/>
12   </inertial>
13   <collision name="collision">
14     <geometry>
15       <mesh filename="package://usv_sim/meshes/simpleHull3/Wing_Eppler479.dae" scale="0.006
16         0.006 0.006"/>
17     </geometry>
18     <origin rpy="1.570796327 0 {PI}" xyz="0.10 0 0.3"/>
19   </collision>
20   <visual name="visual">
21     <geometry>
22       <mesh filename="package://usv_sim/meshes/simpleHull3/Wing_Eppler479.dae" scale="0.006
23         0.006 0.006"/>
24     </geometry>
25     <origin rpy="1.570796327 0 {PI}" xyz="0.10 0 0.3"/>
26   </visual>
27 </link>

```

Listagem 7.2: Exemplo de código das juntas SDF

```

1 <joint name="sail_joint" type="revolute">
2   <parent link="base_link"/>
3   <child link="sail"/>
4   <origin rpy="0 0 0" xyz="0.30 0 0.10"/>
5   <dynamics damping="1.0"/>
6   <axis xyz="0 0 1"/>
7   <limit effort="100" velocity="70" lower="-{PI}" upper="{PI}"/>
8 </joint>

```

SDF é um formato XML que descreve objetos e ambientes para simuladores, visualização e controlo de robôs. Originalmente desenvolvido como parte do simulador de robôs Gazebo. O SDF tornou-se um formato estável, robusto e extensível, capaz de descrever todos os aspetos de robôs, objetos estáticos e dinâmicos, entre outros [95].

URDF é um formato XML usado no ROS para descrever todos os elementos de um robô. Além da inclusão e utilização de todos os parâmetros e *macros* distintos, dentro dos ficheiros URDF também são definidos os parâmetros de propriedades como *links*, *joints*, *visual*, *collision*, *mass*, *inertial*, *pose*, *sensors*, *plugins*, *geometry* e todos os parâmetros necessários para cada *plugin* [96].

Os ficheiros URDF descrevem todas as informações sobre o robô, isso inclui

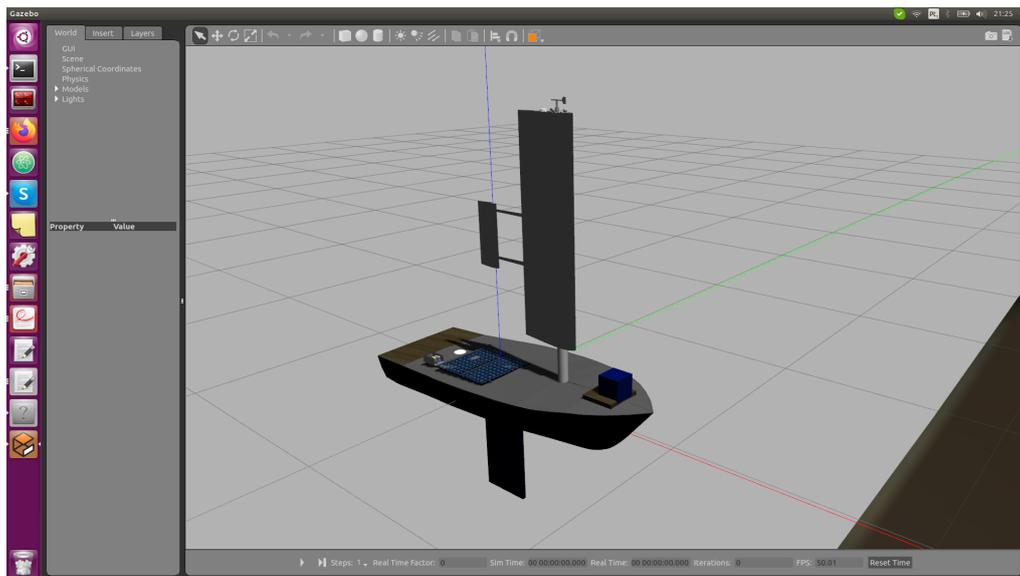


Figura 7.6: Modelo do veleiro no Gazebo.

as suas propriedades materiais, dinâmica, sensores, *plugins*, algoritmos de navegação, entre outros.

O simulador Gazebo utiliza o ficheiro SDF para a descrição do robô, que é semelhante aos ficheiros URDF, mas com informações extras que são exclusivas para o simulador. Dentro dos ficheiros URDF é usado o parâmetro `< gazebo >` para descrever todas as informações sobre o robô que são exclusivas do ficheiro SDF. Antes do robô ser gerado no mundo da simulação, o ficheiro URDF é traduzido automaticamente num ficheiro SDF pelo Gazebo e é carregado no mundo Gazebo.

Xacro é uma linguagem de definição de macros de XML. Permite adicionar novas funcionalidades, incluindo propriedades, blocos de propriedades, expressões matemáticas, blocos de condição, comandos *rospack*, *macros*, parâmetros padrão e a inclusão de outros ficheiros Xacro [97].

7.3.2 Ficheiros

Ficheiro do Mundo O ficheiro do mundo contém toda a descrição sobre os elementos na simulação, incluindo modelos, luzes, sensores e física. O formato do ficheiro do mundo é o mesmo do ficheiro do modelo, utilizando o formato SDF, com extensão `.world`. O servidor Gazebo (*gzserver*) lê esse ficheiro para gerar e preencher o mundo na simulação. Na Listagem 7.3 pode-se observar como é constituído o ficheiro `.world`. Os ficheiros `.world` podem ser personalizados

dentro dos próprios pacotes ROS. Na Listagem 7.4 pode-se observar o ficheiro de lançamento da simulação.

Listagem 7.3: Exemplo de código para o ficheiro do mundo da simulação.

```

1 <?xml version="1.0" ?>
2 <sdf version="1.4">
3   <world name="surface">
4     <include>
5       <uri>model://sun</uri>
6     </include>
7     <wind>
8       <linear_velocity>0 0 0</linear_velocity>
9     </wind>
10    <model name="model">
11      <link name="link">
12        <wind>true</wind>
13      </link>
14    </model>
15  </world>
16 </sdf>

```

Listagem 7.4: Exemplo de código para carregar o mundo.

```

1 <!-- Launch Gazebo with empty world -->
2 <include file="$(find gazebo_ros)/launch/empty_world.launch">
3 <arg name="use_sim_time" value="true" />
4 <arg name="debug" value="false" />
5 <arg name="gui" value="true" />
6 <arg name="paused" value="true"/>
7 <arg name="verbose" value="true"/>
8 <arg name="world_name" value="$(find usv_sim)/world/empty.world"/>
9 </include>

```

Ficheiros do Modelo O ficheiro do modelo recorre ao mesmo formato SDF do ficheiro do mundo, podendo ser gerado diretamente na simulação com o nó `spawn_robot` no ROS. Isso possibilita a utilização de ficheiros Xacro que permitem a gestão e implementação de parâmetros que permitem ao utilizador ajustar os aspetos do robô, sensor ou simulação. O nó `spawn_robot` converte o ficheiro Xacro no ficheiro SDF para interpretação do simulador Gazebo. O método `spawn_robot` utiliza o *script* Python para fazer uma solicitação de chamada de serviço ao nó `gazebo_ros` para adicionar um ficheiro URDF personalizado num ambiente simulado. O *script* está localizado no pacote `gazebo_ros`.

7.4 Controlo

O sistema de controlo para a cauda e para a direção foi implementado num nó ROS denominado `usv_control`, no pacote `usv_control_plugin`. O controlo da cauda utiliza um controlador PD e o controlo da direção recorre ao controlador P. O controlador PD apresenta uma malha de controlo, onde a parte proporcional é obtida multiplicando o erro por uma constante de ganho proporcional e , a parte diferencial é calculada subtraindo o erro atual e o erro anterior, dividindo o resultado pelo tempo decorrido, em seguida, multiplicando pela constante de ganho diferencial. Ambas as partes são adicionadas, resultando num único valor. O controlador P é idêntico e inclui apenas a componente proporcional [98].

7.4.1 Controlo da Vela Asa

O controlo do ângulo de ataque da vela asa é feito através do controlo da cauda. A cauda possui um controlador PD, que possui como entradas a direção do vento aparente, publicada pelo tópico do *plugin anemometer*, a posição angular da vela asa e o ângulo desejado da cauda. Na Figura 7.7 pode-se observar o diagrama de blocos do controlo da cauda, onde e é o erro, $\frac{de}{dt}$ o erro diferencial, entre o ângulo de ataque desejado e o atual, K_p e K_d são os ganhos proporcionais e diferenciais, respectivamente. Na Figura 7.8 pode-se observar o fluxograma do controlo da vela asa.

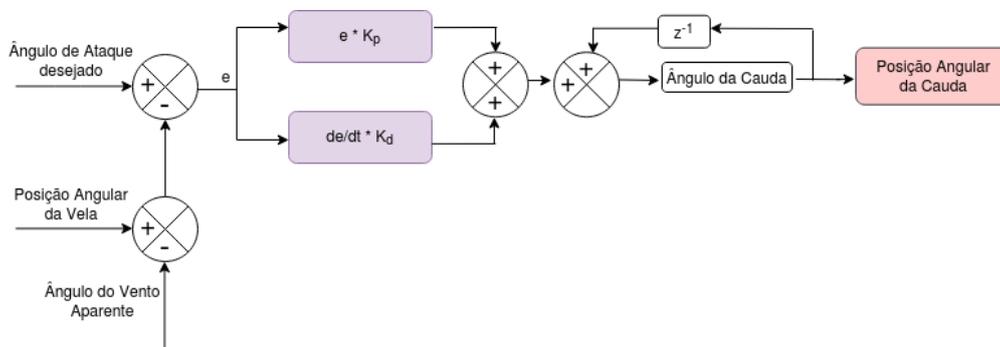


Figura 7.7: Sistema de controlo da vela asa.

7.4.2 Controlo da Direção do Veleiro

O sistema de controlo de direção calcula o erro para o próximo *waypoint*. Se o veleiro estiver próximo do *waypoint*, dado uma margem, este começa a deslocar-se para o *waypoint* seguinte. No entanto, no caso do veleiro ainda estar distante do *waypoint* desejado, a direção do veleiro ($\theta_{desejada}$) é calculada tendo por base o erro anterior e , posteriormente, o erro entre a direção atual do veleiro e a direção

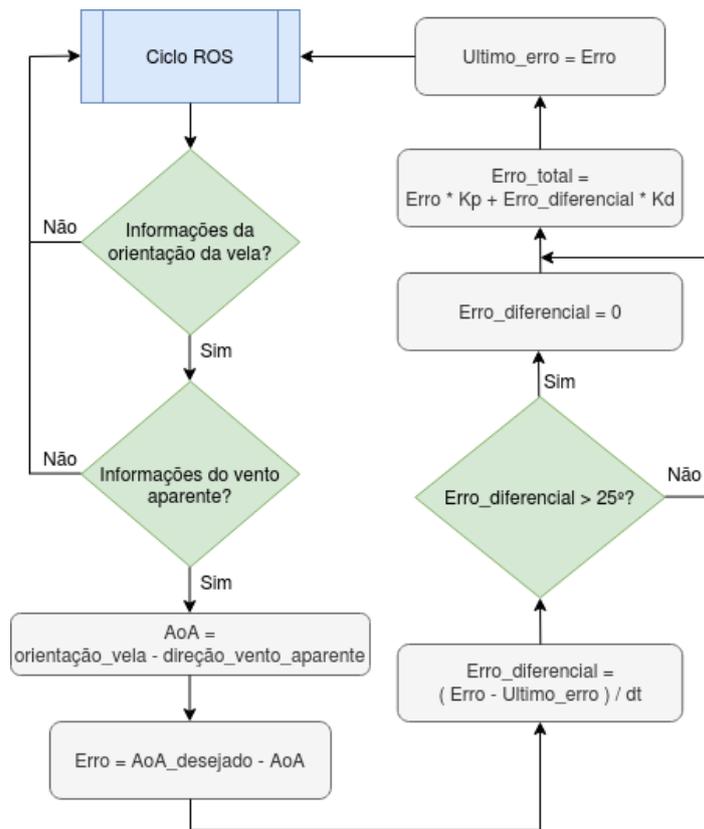


Figura 7.8: Fluxograma do controlo da vela asa.

desejada do veleiro (θ_{erro}). Após o cálculo deste erro, o ângulo de direção é obtido multiplicando o erro pelo ganho P, obtido através da tentativa e erro. Na Figura 7.9 pode-se observar o fluxograma do controlo de direção do veleiro, respetivamente.

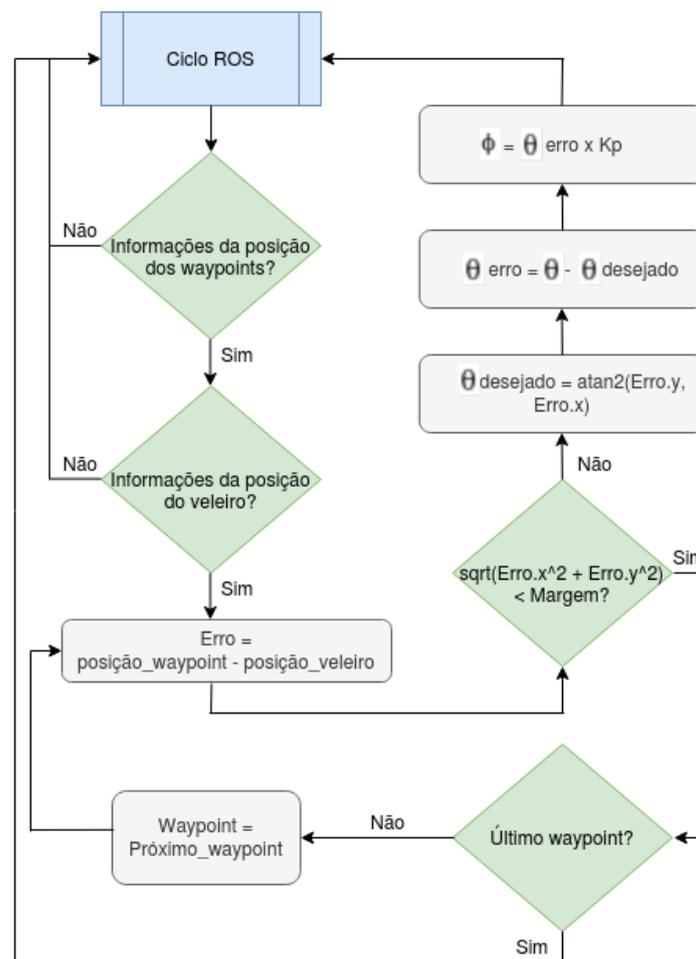


Figura 7.9: Sistema de controlo de direção.

7.5 Conclusão do Capítulo

Este capítulo descreve o processo de simulação, desde a conceção e importação do modelo 3D do veleiro para o ambiente Gazebo até ao controlo do modelo. O sistema de controlo consiste no controlo da cauda e do leme do veleiro. Na cauda utilizou-se um controlador PD para induzir o ângulo de ataque desejado na vela principal. De outro modo, no sistema de controlo do leme utilizou-se um controlador P, responsável por orientar o veleiro na direção pretendida.

Capítulo 8

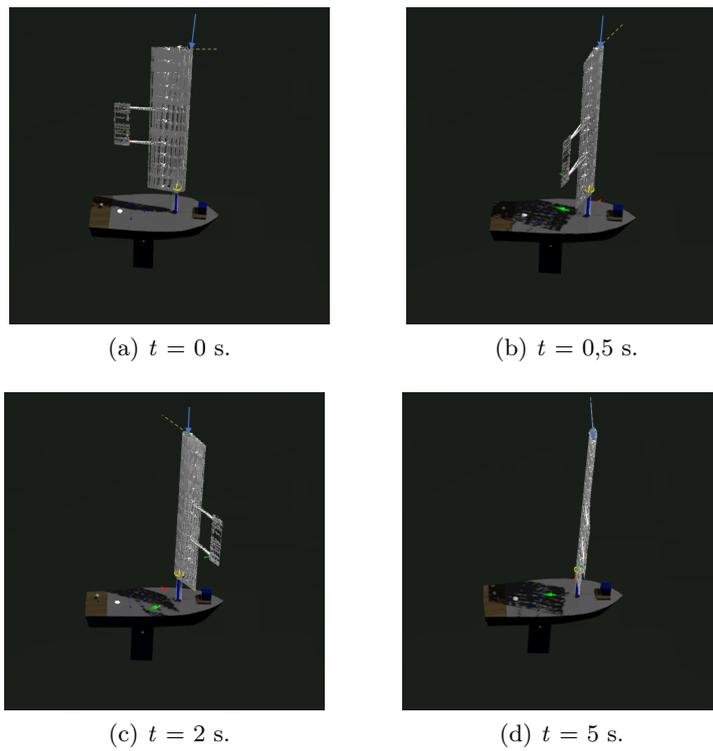
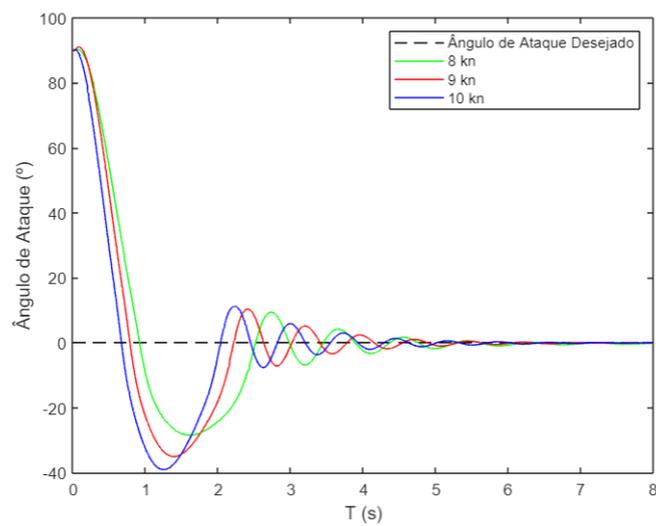
Testes, Resultados e Validações

Neste capítulo são apresentados os testes realizados, os resultados obtidos para validar e caracterizar o veleiro projetado e as respectivas análises. Numa fase inicial o ambiente apresenta diferentes velocidades de vento e direções de vento constantes. Todas as perturbações são negligenciadas.

8.1 Alinhamento da Vela Asa ao Vento

O primeiro teste realizado foi o alinhamento da vela asa ao vento. O veleiro e a vela asa foram orientados de frente para leste (90°). Foi aplicado à simulação um vento vindo de norte (0°), inscrevendo o tópico `pub_wind_topic`, que publica a velocidade e a direção do vento real atual. Essa informação é também subscrita pelo *plugin* do anemómetro que calcula o vento aparente de 90° em relação ao veleiro. Na Figura 8.1 pode-se observar o alinhamento da vela asa ao vento com direção de 0° e velocidade de 8 kn. A seta a azul representa a direção do vento e a linha tracejada a verde a posição da vela asa.

A Figura 8.2 representa a resposta temporal do alinhamento da vela asa com o ângulo de ataque de 0° para as velocidades de vento de 8 kn, 9 kn e 10 kn, a verde, vermelho e azul, respetivamente. A linha tracejada a preto representa o ângulo de ataque desejado. O objetivo destes resultados é demonstrar que, quando a vela asa e a cauda estão alinhadas, a vela asa alinha-se automaticamente ao vento devido ao momento de *pitching* do *plugin* da aerodinâmica. Para diferentes velocidades de vento, a vela asa demora um tempo idêntico a estabilizar. No entanto, verifica-se que, quanto maior a velocidade, maior o *overshoot*, conforme demonstrado na Figura 8.2.

Figura 8.1: Alinhamento da vela asa ao vento com direção de 0° .Figura 8.2: Resposta do alinhamento da vela asa com ângulo de ataque de 0° .

8.2 Controlo da Vela Asa

Com o objetivo de validar o controlo da cauda e a propulsão do veleiro foram efetuados os seguintes testes:

- o teste de orientação da vela asa para o ângulo de ataque definido;
- verificação do funcionamento do anemómetro.

8.2.1 Vela Asa Ajustada através da Cauda

Na realização deste teste o veleiro dirige-se para leste e o vento foi definido pela subscrição ao tópico `pub_wind_topic`. A direção do vento definida foi vindo de norte (0°) e as velocidades do vento definidas foram 8 kn, 9 kn e 10 kn. A cauda foi controlado com dois conjuntos de ganhos PD: $K_p = 0,5$ e $K_d = 0,07$; $K_p = 0,5$ e $K_d = 0,02$ e o ângulo de ataque estabelecido foi $8,5^\circ$. Na Figura 8.3 pode-se observar a vela asa controlada pela cauda para a velocidade de vento de 8 kn. A seta a azul representa a direção do vento e a linha a tracejada a verde a posição da vela asa.

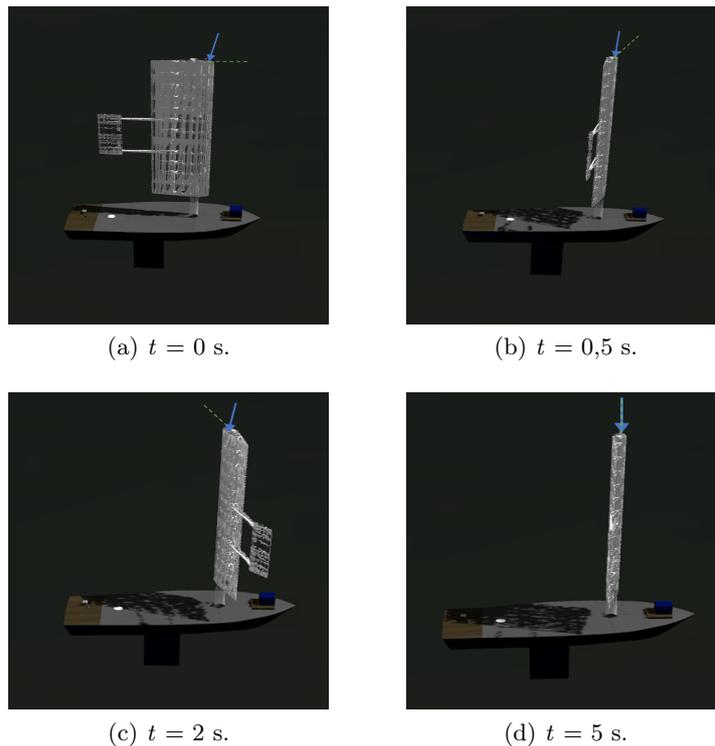


Figura 8.3: Vela asa controlada pela cauda com direção do vento de 0° .

Tendo por base o ângulo de ataque definido, a vela asa perfila-se com a direção do vento. À medida que o erro entre o ângulo de ataque atual e o ângulo de ataque

desejado diminui, o sistema de controlo começa a estabilizar a vela asa para o ângulo de ataque desejado. O tempo que a vela asa leva para atingir o ângulo de ataque desejado é idêntico ao tempo que leva para se alinhar com o vento sem controlo da cauda. Porém, oscila durante mais alguns segundos devido ao ajuste do ângulo da cauda, através do sistema de controlo, de forma à vela asa possuir o ângulo de ataque desejado. Na Figura 8.4 e na Figura 8.5 pode-se observar a resposta temporal do ângulo de ataque da vela asa para os ganhos $K_p = 0,5$ e $K_d = 0,07$; $K_p = 0,5$ e $K_d = 0,02$, respetivamente.

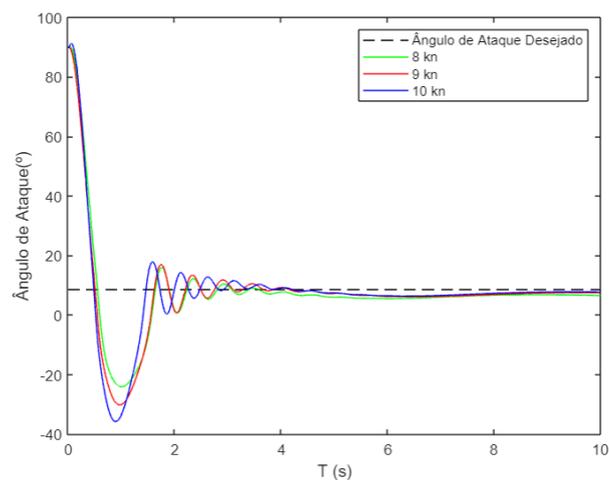


Figura 8.4: Resposta temporal do ângulo de ataque da vela asa com vento de 0° ($K_p = 0,5$, $K_d = 0,07$, $AoA = 8,5^\circ$).

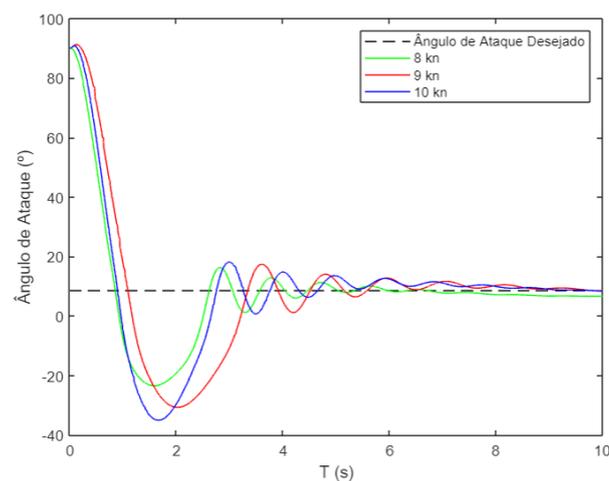
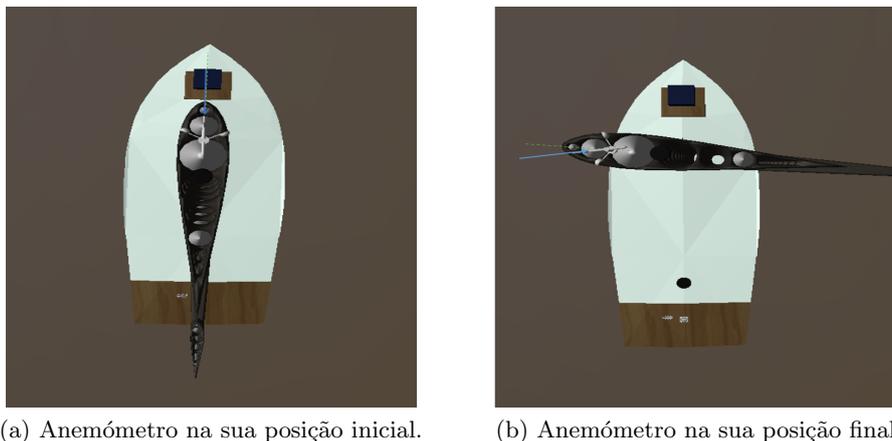


Figura 8.5: Resposta temporal do ângulo de ataque da vela asa com vento de 0° ($K_p = 0,5$, $K_d = 0,02$, $AoA = 8,5^\circ$).

8.2.2 Funcionamento do Anemómetro

O anemómetro, como referido anteriormente, é um sensor de vento que mede a direção e a velocidade do vento aparente. A realização deste teste permite verificar o funcionamento do sensor de vento. O veleiro encontra-se posicionado de frente para leste (90°). Na Figura 8.6a pode-se observar o anemómetro na sua posição inicial (ângulo de ataque de 0°). Na Figura 8.6b pode-se observar a posição do anemómetro quando aplicado à simulação um vento vindo de norte (0°), subscrevendo o tópic `pub_wind_topic`, que publica a velocidade e a direção do vento real atual. O *plugin* do anemómetro subscreve esta informação e calcula o vento aparente em relação ao veleiro. Como dito anteriormente, o ângulo de ataque desejado é de $8,5^\circ$ e, sabendo a direção do vento aparente e a direção da vela asa, foi efetuado o controlo da vela asa de forma a que o ângulo de ataque desejado seja de $8,5^\circ$, como demonstrado na Figura 8.6b. A seta a azul representa a direção do vento aparente e a linha tracejada a verde a posição da vela asa.



(a) Anemómetro na sua posição inicial.

(b) Anemómetro na sua posição final.

Figura 8.6: Verificação do funcionamento do anemómetro com direção do vento de 0° e $AoA = 8,5^\circ$.

8.3 Controlo de Direção

Após a validação do sistema de controlo da vela asa, o controlo de direção do veleiro foi submetido a diversos testes: (i) o veleiro a percorrer um determinado percurso, (ii) o veleiro a percorrer uma determinada área e (iii) o veleiro a contornar obstáculos sujeito a perturbações ambientais. O veleiro foi posicionado inicialmente de frente para leste na posição $P_i = (250;95;1)$ com uma velocidade de vento de 8 kn e direção de 0° . Em relação aos ganhos PD do controlador da cauda foram definidos como $K_p = 0,5$ e $K_d = 0,02$. Relativamente ao ganho P do controlador de direção foi definido como $K_p = 1$.

8.3.1 Varrimento de Percurso

Este teste tinha como objetivo demonstrar que o veleiro era capaz de percorrer um determinado percurso previamente definido, alcançando os pontos de referências desejados. A Figura 8.7 apresenta as capturas de ecrã ilustrativas do percurso, assinalado a cinza, realizado pelo veleiro durante a manobra de *station keeping*.

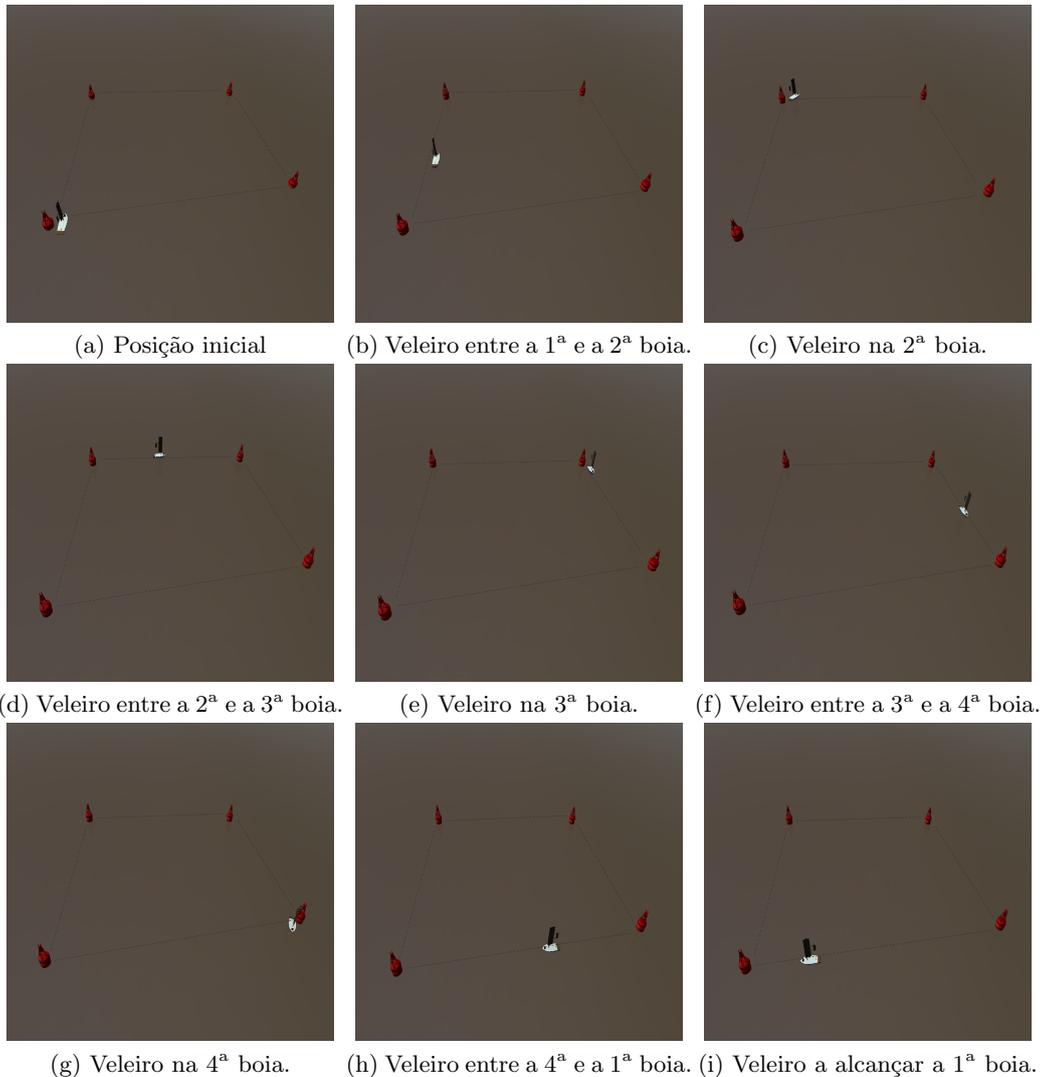


Figura 8.7: Veleiro a percorrer um determinado percurso.

A Figura 8.8 apresenta o perímetro correspondente efetuado.

8.3.2 Varrimento de Área

O segundo teste, apresentado na Figura 8.9, tinha como intenção o varrimento de uma determinada área. Este teste efetua-se quando se pretende analisar uma

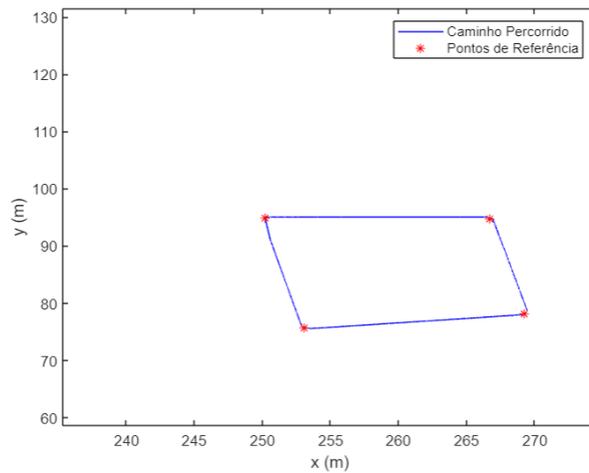


Figura 8.8: Veleiro a percorrer um determinado perímetro.

determinada área e, portanto, realiza-se o varrimento da mesma. Na Figura 8.10 pode-se observar algumas capturas de ecrã do veleiro a varrer uma determinada área.

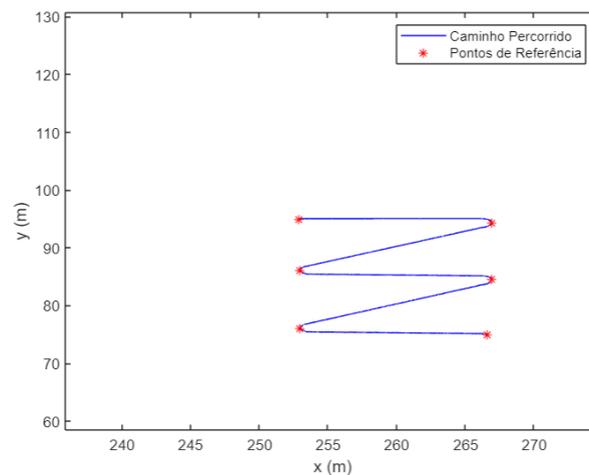


Figura 8.9: Veleiro a varrer uma determinada área.

8.3.3 Contorno de Obstáculos

Por fim, no último teste, o veleiro foi orientado inicialmente para leste na posição $P_i = (240;100;1)$. Este teste surgiu da necessidade de analisar o comportamento do veleiro sujeito a uma perturbação ambiental, neste caso, ondulação. E demonstrar que este era capaz de contornar os obstáculos e prosseguir com a sua trajetória. Na Figura 8.11 pode-se observar o caminho percorrido pelo veleiro

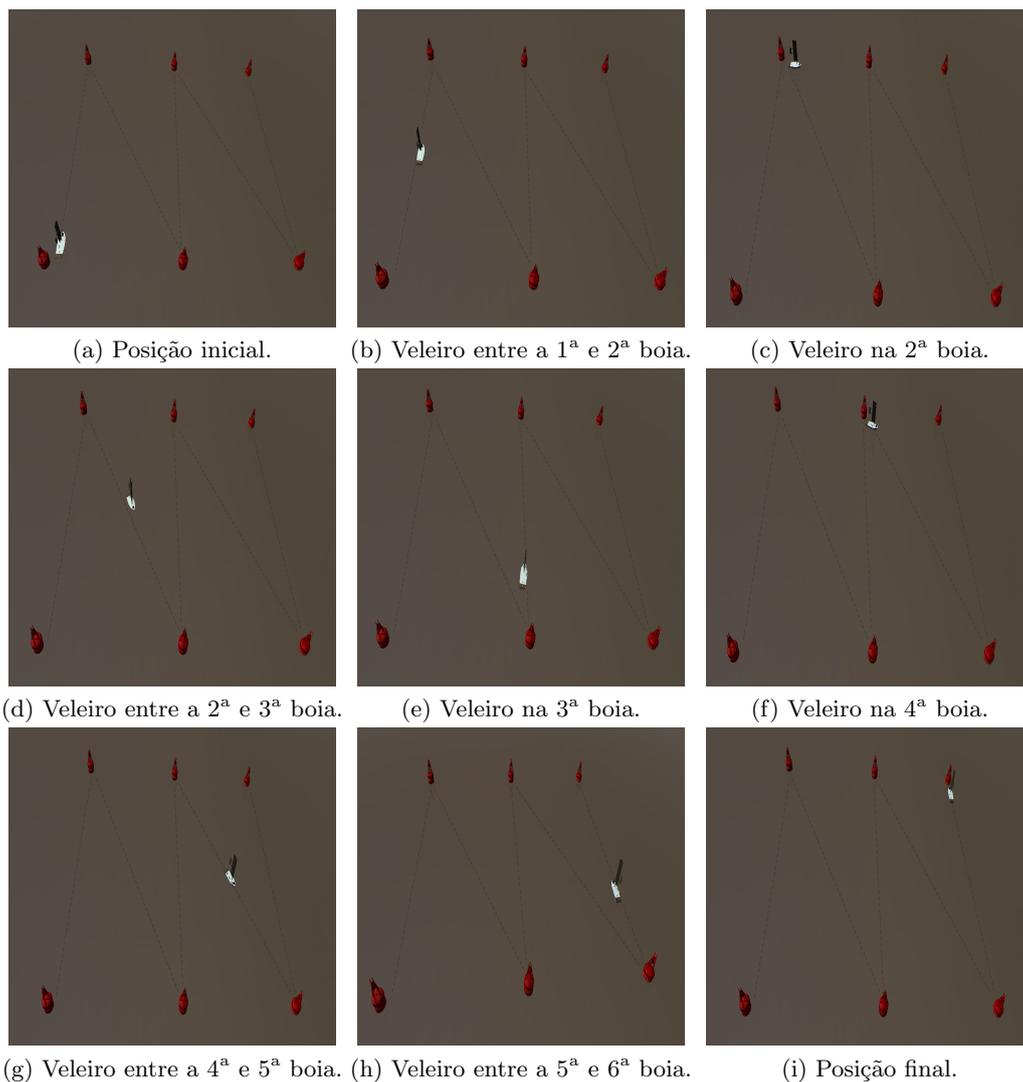


Figura 8.10: Varrimento de uma área.

sem perturbações e com perturbações. Na Figura 8.12 pode-se observar algumas capturas de ecrã do veleiro a desviar-se dos obstáculos, sem perturbações e com perturbações.

8.4 Diagrama Polar

O diagrama polar caracteriza o desempenho óptimo de uma plataforma propulsionada pelo vento em função da velocidade do vento e do ângulo do vento aparente. O veleiro foi exposto a três testes, onde respetivamente a velocidade do vento foi 8 kn, 9 kn e 10 kn. Para cada velocidade do vento, a direção do vento foi alterada em intervalos de 10° de 0° a $\pm 180^\circ$. Relativamente ao controlo autónomo da

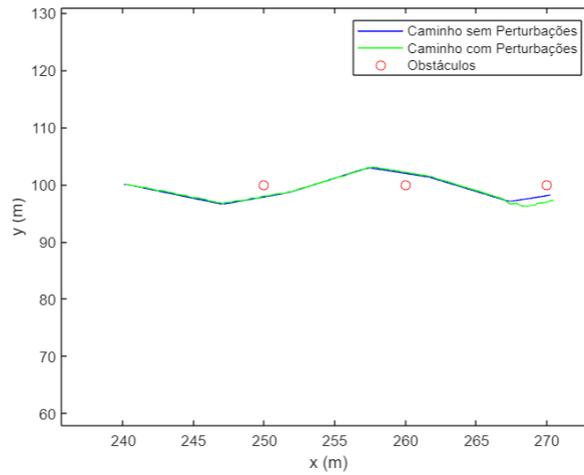
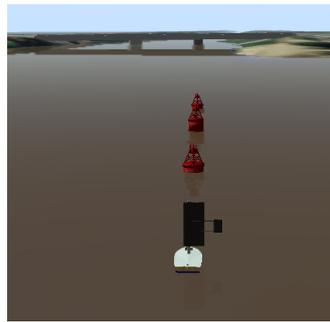


Figura 8.11: Veleiro a contornar os obstáculos sem e com perturbações.

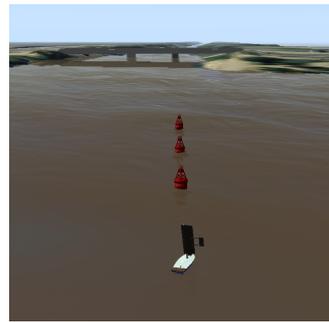
cauda, os ganhos do controlador PD foram definidos como $K_p = 0,5$ e $K_d = 0,02$.

Na Figura 8.13 pode-se observar a velocidade do veleiro para as diferentes direções do vento. A velocidade do veleiro começou a aumentar aproximadamente aos $\pm 35^\circ$ até alcançar o pico em $\pm 90^\circ$. Depois do pico, a velocidade máxima do veleiro começou a diminuir até atingir valores nulos próximos de $\pm 180^\circ$. A zona proibida para navegar é visível entre -35° e $+35^\circ$ e entre -135° e $+135^\circ$, conforme demonstrado no diagrama polar na Figura 8.13. Os gráficos a verde, vermelho e azul representam o comportamento do veleiro para a velocidade de vento de 8 kn, 9 kn e 10 kn, respetivamente.

O principal objetivo da realização do diagrama polar é analisar o comportamento dinâmico do veleiro em todas as direções do vento.



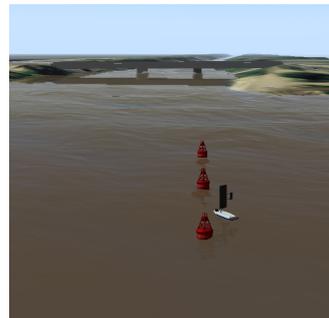
(a) Posição inicial.



(b) Posição inicial.



(c) Contorno da primeira bóia.



(d) Contorno da primeira bóia com perturbações.



(e) Contorno da segunda bóia.



(f) Contorno da segunda bóia com perturbações.

Figura 8.12: Contorno dos obstáculos sem perturbações (lado esquerdo) e com perturbações do mar (lado direito).

8.5 Conclusão do Capítulo

Este capítulo apresenta os testes efetuados para validar o modelo e respetivo controlo.

Relativamente ao alinhamento da vela asa ao vento verificou-se que a vela asa e a cauda alinham-se ao vento, sendo o ângulo de ataque 0° conforme demonstrado na resposta temporal da Figura 8.2. Em relação ao controlo da vela asa, verificou-

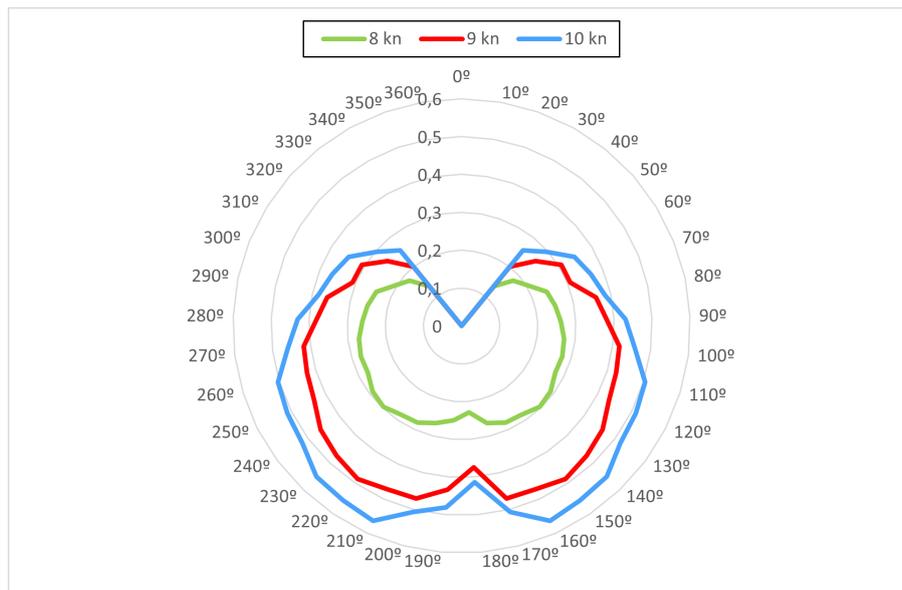


Figura 8.13: Diagrama polar.

se que a vela asa orienta-se para o ângulo de ataque definido. E, o anemómetro funciona corretamente.

Em relação ao controlo de direção do veleiro, elaborou-se três experiências. A primeira experiência diz respeito ao varrimento de percurso e, verificou-se que o veleiro é capaz de percorrer um determinado perímetro previamente definido. A segunda experiência tem como finalidade varrer uma área. Por fim, a última experiência realizada foi o contorno de obstáculos com e sem ondulação do mar, com o intuito de verificar se o veleiro era capaz de contornar possíveis obstáculos que possam surgir no mar. Além disso, foi elaborado o diagrama polar que permite analisar o comportamento dinâmico do veleiro em todas as direções do vento.

Em suma, concluiu-se que o controlo da vela asa funciona corretamente. Assim como o controlo de direção do veleiro, permitindo alcançar os pontos de referência desejados. Com base nos resultados obtidos, concluiu-se que o simulador desenvolvido fornece um modelo adequado e garante estabilidade e controlo preciso para fins de exploração.

Capítulo 9

Conclusões

Neste capítulo são apresentados os resultados obtidos, as dificuldades que surgiram e sugestões de melhorias de trabalho.

9.1 Resultados

Esta dissertação aborda o projeto e simulação de um veleiro autónomo. A plataforma proposta é um veleiro com controlo de direção e uma vela asa rígida, controlada por uma cauda. O modelo da vela asa e da cauda, baseado no perfil EPPLER 479, foram concebidos no Autodesk Fusion 360 e importados para o Gazebo. A vela asa apresenta uma área de $1,88 \text{ m}^2$, com envergadura de 2,50 m e comprimento de corda de 0,75 m. A cauda apresenta uma área de $0,23 \text{ m}^2$, com envergadura de 0,75 m e comprimento de corda de 0,30 m. O controlo do veleiro compreende o controlador da cauda, que aciona a vela asa com base no ângulo de ataque atual e no ângulo de ataque desejado. E, o controlo de direção com base na posição atual e desejada do veleiro. O testes foram realizados recorrendo ao ROS e ao Gazebo, juntamente com três *plugins* de código aberto reutilizáveis, o *plugin* aerodinâmico, o *plugin* da flutuabilidade e o *plugin* dos sensores e três *plugins* criados, o *plugin* que estabelece a interface do ROS com as juntas da cauda e do leme, o *plugin* do cálculo do vento aparente e o *plugin* do controlo da plataforma.

9.2 Dificuldades

No desenvolvimento desta dissertação surgiram diversas dificuldades, como o funcionamento do simulador Gazebo, devido à pouca documentação disponível, embora houvesse alguma familiaridade com o simulador. Além disso, o objetivo

original da dissertação foi alterado para um ambiente simulado, devido à pandemia mundial.

Uma das grandes dificuldades desta dissertação foi a introdução de perturbações no ambiente Gazebo. As correntes marítimas eram uma das perturbações que se pretendia implementar, no entanto não foi possível obter esse ficheiro por falta de informação. A tentativa de contactar os autores do USVSim foi feita, no entanto sem sucesso. Contudo, foi possível simular a ondulação do mar através da configuração do ficheiro XML presente na pasta `scene` do UWSim.

9.3 Trabalho Futuro

Uma das melhorias que podem ser consideradas é a construção de um diagrama polar com menor diferença nos ângulos do vento. A direção do vento foi alterada em intervalos de 10° , o que resultou numa aproximação para os pontos intermédios. A direção ideal do vento seria incrementos de 1° , uma vez que forneceria um diagrama polar mais detalhado, no entanto exigiria mais tempo de processamento. Além disso, também seria interessante testar para mais velocidades de vento.

Outra possível melhoria seria experimentar diferentes perfis alares para o modelo da vela asa, assim como a alteração da envergadura e da corda da vela asa. A modificação do perfil alar teria um grande impacto na dinâmica do modelo do veleiro porque as características de sustentação e arrasto alterar-se-iam assim como a configuração do ângulo de ataque. O teste consiste em alterar algumas características importantes de forma a comparar os diferentes modelos. Dado que o sistema de controlo da vela asa é independente da vela, este pode ser aplicado para qualquer tipo de modelo de vela asa.

Além disso, considera-se fundamental o desenvolvimento de um estimador não linear, como um Filtro de Kalman Extended ou Filtro de Partículas, com o objetivo de estimar o estado do veleiro, nomeadamente a sua posição e velocidade recorrendo aos dados dos sensores.

E, por fim, a construção, desenvolvimento e avaliação da versão real do veleiro projetado é um dos trabalhos futuros a considerar.

9.4 Conclusão do Capítulo

O desenvolvimento desta dissertação contribuiu com a elaboração de um modelo de veleiro autónomo e com a criação e aprimoramento de *plugins* Gazebo que podem ser reutilizados noutras plataformas robóticas.

Tendo por base os objetivos do Capítulo 1 pode-se concluir que a solução desenvolvida foi projetada e simulada com sucesso.

O controlo da cauda foi capaz de orientar a vela asa para o ângulo de ataque desejado, em segundos, para as várias velocidades de vento, tornando esta situação ideal para uma plataforma equipada com vela asa. O controlo de direção do veleiro permitiu elaborar os diversos testes com sucesso, sendo o veleiro capaz de percorrer um determinado percurso assim como desviar-se de obstáculos.

Além disso, devido ao distanciamento social e ao confinamento domiciliar durante a pandemia, esta dissertação ajudou a melhorar as habilidades sociais, como organização e gestão do tempo, que se mostraram cruciais.

Bibliografia

- [1] C. Sauzé and M. Neal, “Design considerations for sailing robots performing long term autonomous oceanography,” in *Proceedings of the International Robotic Sailing*, 2008. Acedido em março de 2020. [citado na p. vii, 9, 10, 11, 12, 13, 23, 24, 27]
- [2] C. Sauzé, “A Neuro-Endocrine Inspired Approach to Power Management in Sailing Robots,” Master’s thesis, The University of Wales, 2010. Acedido em março de 2020. [citado na p. vii, 9, 10, 12, 14, 15, 23, 24, 25]
- [3] M. Silva, A. Friebe, B. Malheiro, P. Guedes, P. Ferreira, and M. Waller, “Rigid wing sailboats: A state of the art survey,” in *Ocean Engineering*, 2019. Acedido em março de 2020. [citado na p. vii, 9, 15, 18, 19, 20, 37]
- [4] A. C. Domínguez, B. Valle-Fernández, J. Cabrera-Gámez, A. R. de Miguel, and J. C. García, “A-TIRMA G2: An Oceanic Autonomous Sailboat,” in *Proceedings of the 8th International Robotic Sailing Conference*, 2015. Acedido em março de 2020. [citado na p. vii, viii, 15, 16, 17, 35, 36]
- [5] U. Dhomé, “Further development and performance evaluation of the autonomous sailing boat Maribot Vane,” Master’s thesis, Royal Institute of Technology, 2017. Acedido em março de 2020. [citado na p. vii, 18, 19]
- [6] Microtransat, “The Microtransat Challenge.” <https://www.microtransat.org/>, 2021. Acedido em março de 2020. [citado na p. vii, 20, 24, 30]
- [7] Offshore Sensing - Sailbuoy, “Transatlantic crossing 2018.” <http://www.sailbuoy.no/news-2/64-transatlantic-crossing>, 2021. Acedido em abril de 2020. [citado na p. vii, 21]
- [8] Aberystwyth University, “Project: Pinta.” <https://www.aber.ac.uk/en/cs/research/ir/robots/pinta/>, 2021. Acedido em março de 2020. [citado na p. vii, 23]

- [9] H. Erckens, G. Beusser, C. Pradalier, and R. Y. Siegwart, “Avalon,” *IEEE Robotics & Automation Magazine*, vol. 17, no. 1, pp. 45–54, 2010. [citado na p. vii, 25]
- [10] Inflated Wing Sails, “Inflated Wing Sails.” <http://inflatedwingsails.com/concept/>, 2021. Acedido em maio de 2020. [citado na p. vii, 28, 29]
- [11] AviationChief, “Angle of Attack.” <http://www.aviationchief.com/angle-of-attack.html>, 2021. Acedido em junho de 2020. [citado na p. viii, 58]
- [12] Airfoil Tools, “EPPLER 479 AIRFOIL (e479-il).” <http://airfoiltools.com/airfoil/details?airfoil=e479-il>, 2021. Acedido em junho de 2020. [citado na p. viii, 59]
- [13] M. Silva, B. Malheiro, P. Guedes, and P. Ferreira, “Airfoil Selection and Wingsail Design for an Autonomous Sailboat,” in *Robot 2019: Fourth Iberian Robotics Conference*, 2019. Acedido em junho de 2020. [citado na p. viii, 56, 58, 60, 61, 62, 63, 64, 65, 67]
- [14] Kama Yachts, “PROTON 2.4 mR -technical data.” <http://kamayachts.com/en-proton.html>, 2021. Acedido em junho de 2020. [citado na p. viii, 65, 66]
- [15] M. Prats, J. Pérez, J. J. Fernández, and P. J. Sanz, “An Open Source Tool for Simulation and Supervision of Underwater Intervention Missions,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012. Acedido em agosto de 2020. [citado na p. viii, 70]
- [16] R. Mendonça, P. Santana, F. Marques, A. Lourenço, J. Silva, and J. Barata, “Kelpie: A ros-based multi-robot simulator for water surface and aerial vehicles,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3645–3650, Oct 2013. [citado na p. viii, 71, 72]
- [17] M. Paravisi, D. Santos, and A. Amory, “Unmanned Surface Vehicle simulation on Gazebo with water current and winds.” https://github.com/disaster-robotics-proalertas/usv_sim_lsa, 2019. Acedido em agosto de 2020. [citado na p. viii, 73, 74, 75]
- [18] A. Koubaa, “Robot Operatins System (ROS The Complete Reference (Volume 1),” in *Studies in Computational Intelligence 625*, 2016. Acedido em julho de 2020. [citado na p. viii, 81]
- [19] Gazebo, “Tutorial: Ros integration overview.” http://gazebosim.org/tutorials?tut=ros_overview, 2014. Acedido em dezembro de 2020. [citado na p. viii, 84]

- [20] M. F. S. Vitor Tinoco, Benedita Malheiro, “Design, Modelling and Simulation of a Wing SailLand Yacht,” *Applied Sciences*, 2021. Acedido em julho de 2021. [citado na p. viii, 86]
- [21] Gazebo, “Aerodynamics.” <http://gazebosim.org/tutorials?tut=aerodynamics&cat=physics>, 2014. Acedido em agosto de 2020. [citado na p. viii, 85, 86, 87]
- [22] NautiCurso, “Detalhes das partes de um veleiro.” <https://www.nauticurso.com.br/partes-veleiros.html>, 2013. Acedido em abril de 2020. [citado na p. ix]
- [23] S. Jollands, “Points of Sailing.” <https://safe-skipper.com/points-of-sail/>, 2021. Acedido em maio de 2020. [citado na p. 6]
- [24] N. Ramos, “Mundo da Vela: Rumo e Mareações.” <https://sites.google.com/site/marvelavento/lingua-language/portugues/conceitos/rumo-e-mareacoes>, 2013. Acedido em maio de 2020. [citado na p. 6]
- [25] Bowditch, *The American Practical Navigator*. Paradise Cay Publications, 2010. Acedido em março de 2020. [citado na p. 7]
- [26] N. A. Cruz and J. C. Alves, “Ocean sampling and surveillance using autonomous sailboats,” in *International Robotic Sailing Conference 2008*, 2008. Acedido em abril de 2020. [citado na p. 7]
- [27] R. Stelzer and K. Jafarmadar, “History and Recent Developments in Robotic Sailing,” *Reliable Boat for Crossing the Atlantic Ocean*, pp. 3–23, 2011. Acedido em março de 2020. [citado na p. 7, 12, 24]
- [28] G. H. Elkaim, “System Identification for Precision Control of a WingSailed GPS-Guided Catamaran,” Master’s thesis, Stanford University, 2001. Acedido em março de 2020. [citado na p. 8, 9, 27, 61, 63, 88]
- [29] C. Sauzé and M. Neal, “An Autonomous Sailing Robot for Ocean Observation,” in *Proceedings of the 7th Towards Autonomous Robotic Systems (TAROS) Conference*, 2006. Acedido em março de 2020. [citado na p. 9, 11, 12, 27, 34]
- [30] Aberystwyth University, “Project: ARC.” <https://www.aber.ac.uk/en/cs/research/ir/robots/arc/>, 2021. Acedido em março de 2020. [citado na p. 10, 11]
- [31] Kompai Robotics, “Kompai Robotics.” <https://kompairobotics.com/>, 2021. Acedido em março de 2020. [citado na p. 12]

- [32] C. Sauzé and M. Neal, “Long term autonomous robotic sailing,” in *Proceedings of the 4th International Robotic Sailing Conference*, pp. 113–124, 2011. Acedido em março de 2020. [citado na p. 12]
- [33] A. Friebe, M. Olsson, M. le Gallic, J. L. Springett, K. Dahl, and M. Waller, “A Marine Research ASV Utilizing Wind and Solar Power,” in *OCEANS 2017 - Aberdeen*, 2017. Acedido em março de 2020. [citado na p. 17, 18]
- [34] Offshore Sensing - Sailbuoy, “SailBuoy - Unmanned Surface Vessel.” <http://www.sailbuoy.no/>, 2021. Acedido em março de 2020. [citado na p. 19]
- [35] Tech Xplore, “Robot boat sails into history by finishing Atlantic crossing.” <https://techxplore.com/news/2018-09-robot-boat-history-finishing-atlantic.html>, 2018. Acedido em Abril de 2020. [citado na p. 20]
- [36] INESC TEC, “FEUP/INESC TEC Robotic Sailboat Wins International Competition.” <https://www.inesctec.pt/en/news/feup-inesc-tec-robotic-sailboat-wins-international-competition-17849#intro>, 2012. Acedido em março de 2020. [citado na p. 22, 23]
- [37] M. Neal, C. Sauzé, B. Thomas, and J. C. Alves, “Technologies for Autonomous Sailing: Wings and Wind Sensors,” in *Proceedings of the 2nd International Robotic Sailing Conference*, pp. 23–30, 2009. Acedido em abril de 2020. [citado na p. 26, 27, 36]
- [38] R. Stelzer and D. E. Dalmau, “A Study on Potential Energy Savings by the Use of a Balanced Rig on a Robotic Sailing Boat,” in *Robotic Sailing 2012 – Proceedings of the 5th International Robotic Sailing Conference.*, pp. 87–93, 2012. Acedido em abril de 2020. [citado na p. 27, 36, 44]
- [39] Robotic Sailing, “World Robotic Sailing Championship and International Robotic Sailing Conference.” <https://www.roboticsailing.org/>, 2018. Acedido em março de 2020. [citado na p. 30]
- [40] Faculdade Engenharia do Porto, “World Robotic Sailing Championship 2016.” https://web.fe.up.pt/~jca/wrsc2016.com/docs/WRSC_rules_2016_V2.1.pdf, R. Dr. Roberto Frias, 2016. Acedido em março de 2020. [citado na p. 30]
- [41] SailBot, “SailBot International Robotic Sailing Regatta.” <https://www.sailbot.org/>, 2017. Acedido em março de 2020. [citado na p. 31]
- [42] Diprofiber, “Fibras de vidro.” <https://diprofiber.com.br/fibra-de-vidro/>, 2020. Acedido em junho de 2021. [citado na p. 33]
- [43] Toda Matéria, “Aço inox.” <https://www.todamateria.com.br/aco-inox/>, 2011. Acedido em junho de 2021. [citado na p. 34]

- [44] AZO Materials, “Aluminium: Specifications, properties, classifications and classes.” <https://www.azom.com/article.aspx?ArticleID=2863>, 2005. Acedido em junho de 2021. [citado na p. 34]
- [45] Total Materia, “Propriedades do alumínio.” <https://www.totalmateria.com/page.aspx?ID=PropriedadesdoAluminio&LN=PT>, 2021. Acedido em junho de 2021. [citado na p. 34]
- [46] Wikipedia, “Madeira.” <https://pt.wikipedia.org/wiki/Madeira>, 2021. Acedido em julho de 2021. [citado na p. 34]
- [47] General Iron Fittings, “Espécies de madeiras brasileiras.” <https://ironfittings.com.br/especies-de-madeiras-brasileiras/>, 2021. Acedido em julho de 2021. [citado na p. 34]
- [48] Clip Carbono, “Materiais empregados - fibra de carbono.” <https://www.clipcarbono.com/pt/content/10-informacion-variada-a-cerca-de-la-fibra-de-carbono>, 2021. Acedido em junho de 2021. [citado na p. 34]
- [49] Wikipedia, “Patilhão.” <https://pt.wikipedia.org/wiki/Patilhão>, 2020. Acedido em julho de 2021. [citado na p. 36, 131]
- [50] Wikipédia, “Mastro.” <https://pt.wikipedia.org/wiki/Mastro>, 2020. Acedido em junho de 2020. [citado na p. 37]
- [51] PCE Instruments, “Wind speed sensor pce-fst-200-201.” <https://www.industrial-needs.com/technical-data/wind-speed-sensor-pce-fst-200-201.htm>, 2021. Acedido em maio de 2021. [citado na p. 39]
- [52] PCE Instruments, “Wind direction sensor pce-fst-200-202.” <https://www.industrial-needs.com/technical-data/wind-direction-sensor-pce-fst-200-202.htm>, 2021. Acedido em maio de 2021. [citado na p. 39]
- [53] RIKA, “Rk120-01c combined wind speed & direction sensor.” <https://www.rikasensor.com/rk120-01c-combined-wind-speed-direction-sensor-1.html>, 2019. Acedido em maio de 2021. [citado na p. 39, 56]
- [54] Sparton, “What is an imu?.” <https://www.spartonnavex.com/imu/>, 2021. Acedido em maio de 2021. [citado na p. 39, 41]
- [55] SBG Systems, “Ellipse 2 micro series.” <https://www.sbg-systems.com/products/ellipse-micro-series/>, 2021. Acedido em junho de 2021. [citado na p. 41]

- [56] Xsens, “Mti-610 imu.” <https://mtidocs.xsens.com/sensor-specifications-2>, 2021. Acedido em junho de 2021. [citado na p. 41]
- [57] x-io Technologies, “Ngimu.” <https://x-io.co.uk/ngimu/>, 2021. Acedido em junho de 2021. [citado na p. 41]
- [58] TDK Corporation, “Mpu-6000.” https://product.tdk.com/en/search/sensor/motion-inertial/imu/info?part_no=MPU-6000, 2021. Acedido em junho de 2021. [citado na p. 41]
- [59] Bosch Sensortec, “Bmf055.” <https://www.bosch-sensortec.com/products/smart-sensors/bmf055/>, 2021. Acedido em junho de 2021. [citado na p. 41, 56]
- [60] Symmetry Electronics, “What is the difference between gnss and gps?” <https://www.semiconductorstore.com/blog/2015/What-is-the-Difference-Between-GNSS-and-GPS/1550/>, 2021. Acedido em maio de 2021. [citado na p. 41]
- [61] u-blox, “Neo-m8.” <https://www.u-blox.com/en/docs/UBX-15031086>, 2021. Acedido em junho de 2021. [citado na p. 41, 56]
- [62] Banggood, “Beitian bn-880 flight control gps module dual module compass with cable for rc drone fpv racing.” <https://www.banggood.com/Beitian-BN-880-Flight-Control-GPS-Module-Dual-Module-Compass-With-Cable-for-RC-Drone-FPV-Racing-p-971082.html>, 2006. Acedido em junho de 2021. [citado na p. 41]
- [63] VectorNav, “Vn-200 gnss/ins.” https://www.vectornav.com/docs/default-source/datasheets/vn-200-datasheet.pdf?sfvrsn=e1a7b2a0_5, 2021. Acedido em junho de 2021. [citado na p. 41]
- [64] Society of Robots, “Sensors - robot encoder (slot, rotary, linear).” https://www.societyofrobots.com/sensors_encoder.shtml, 2021. Acedido em junho de 2021. [citado na p. 41]
- [65] P. H. Miller, M. Hamlet, and J. Rossman, “Continuous improvements to usna sailbots for inshore racing and offshore voyaging,” in *Robotic Sailing 2012 – Proceedings of the 5th International Robotic Sailing Conference*, pp. 49–60, 2012. Acedido em junho de 2020. [citado na p. 41, 43, 64, 65]
- [66] C. Schröder, L. Hertel, and M. Hamlet, “Development of a Low-Budget Robotic Sailboat,” in *Robotic Sailing 2013 – Proceedings of the 6th International Robotic Sailing Conference*, pp. 13–24, 2013. Acedido em junho de 2020. [citado na p. 43]

- [67] ElectroFun, “Servo motor sg90 9g tower pro.” <https://www.electrofun.pt/robotica/servo-motor-sg90>, 2021. Acedido em junho de 2021. [citado na p. 43]
- [68] TME, “Ser0006 dfrobot.” <https://www.tme.eu/pt/details/df-ser0006/servomotores/dfrobot/ser0006/>, 2021. Acedido em junho de 2021. [citado na p. 43, 51, 55]
- [69] botnroll, “Servo standard 15kg 270° c/ engrenagens metálicas (dss-m15s).” <https://www.botnroll.com/pt/servos/2361-servo-standard-15kg-270-c-engrenagens-metalicas-dss-m15s-.html>, 2021. Acedido em junho de 2021. [citado na p. 43, 51, 55]
- [70] RC Race-Shop, “Cs-5226 hv high torque servo titan waterproof.” <https://www.rc-race-shop.de/en/corally/52001-cs-5226-hv-high-torque-servo-titan-waterproof>, 2021. Acedido em junho de 2021. [citado na p. 43]
- [71] P. Miller, O. M. Brooks, and M. Hamlet, “Development of the usna sailbots (asv),” in *Proceedings of the 2nd International Robotic Sailing Conference*, pp. 9–16, 2009. Acedido em junho de 2020. [citado na p. 44]
- [72] Wikipedia, “Single-board computer.” https://en.wikipedia.org/wiki/Single-board_computer, 2021. Acedido em maio de 2021. [citado na p. 46]
- [73] Social Compare, “Raspberrypi models comparison.” <https://socialcompare.com/en/comparison/raspberrypi-models-comparison>, 2021. Acedido em junho de 2021. [citado na p. 46]
- [74] Seeed Studio, “2km long range rf link kits w/ encoder and decoder.” https://wiki.seeedstudio.com/2KM_Long_Range_RF_link_kits_w_encoder_and_decoder/, 2008. Acedido em maio de 2021. [citado na p. 48]
- [75] DFRobot, “Apc220 radio data module sku tel0005.” https://wiki.dfrobot.com/APC220_Radio_Data_Module_SKU_TEL0005_, 2021. Acedido em maio de 2021. [citado na p. 48, 52]
- [76] Battery University, “Bu-706: Summary of do’s and don’ts.” https://batteryuniversity.com/learn/article/do_and_dont_battery_table, 2021. Acedido em maio de 2021. [citado na p. 49]
- [77] P. Miller, A. Beeler, B. Cayaban, M. Dalton, C. Fach, C. Link, J. MacArthur, J. Urmenita, and R. Y. Medina, “An easy-to-build, low-cost, high-performance sailbot,” in *Robotic Sailing 2014 – Proceedings of the 7th International Robotic Sailing Conference*, pp. 3–16, 2014. Acedido em junho de 2020. [citado na p. 64]

- [78] M. Paravisi, D. H. Santos, V. Jorge, G. Heck, L. M. Gonçalves, and A. Amory, “Unmanned Surface Vehicle Simulator with Realistic Environmental Disturbances,” in *Sensors*, 2019. Acedido em julho de 2020. [citado na p. 69, 72]
- [79] M. Prats, “uwsim.” <http://wiki.ros.org/uwsim>, 2013. Acedido em agosto de 2020. [citado na p. 70]
- [80] Coppelia Robotics, “Coppeliasim.” <https://www.coppeliarobotics.com/>, 2021. Acedido em outubro de 2020. [citado na p. 71]
- [81] Github, “MORSE: the Modular Open Robots Simulator Engine.” <https://github.com/morse-simulator/morse>, 2021. Acedido em julho de 2020. [citado na p. 72]
- [82] Github, “Modular OpenRobots Simulation Engine.” <https://morse-simulator.github.io/>, 2021. Acedido em julho de 2020. [citado na p. 72]
- [83] LinuxLinks, “Gazebo – robot simulation made easy.” <https://www.linuxlinks.com/gazebo-robot-simulation-made-easy/>, 2021. Acedido em julho de 2020. [citado na p. 73]
- [84] N. Koenig and A. Howard, “Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2149–2154, 2004. Acedido em dezembro de 2020. [citado na p. 75]
- [85] R. J. YoonSeok, HanCheol Cho and T. Lim, *ROS Robot Programming*. ROBOTICS Co.,Ltd., 2017. Acedido em agosto de 2020. [citado na p. 80]
- [86] Gazebo, “Gazebo.” <http://gazebosim.org/>, 2014. Acedido em junho de 2021. [citado na p. 83]
- [87] I. Saito, “gazebo ros pkgs.” http://wiki.ros.org/gazebo_ros_pkgs, 2021. Acedido em junho de 2021. [citado na p. 83]
- [88] Gazebo, “Overview of Gazebo Plugins.” http://gazebosim.org/tutorials/?tut=plugins_hello_world, 2014. Acedido em julho de 2020. [citado na p. 84, 85]
- [89] Gazebo, “Gazebo plugins in ROS.” http://gazebosim.org/tutorials?tut=ros_gzplugins, 2014. Acedido em julho de 2020. [citado na p. 85]
- [90] S. Kohlbrecher and J. Meyer, “hector gazebo plugins.” http://wiki.ros.org/hector_gazebo_plugins, 2016. Acedido em maio de 2021. [citado na p. 85]

- [91] J. Smets, “Simulation of Autonomous Wing Sailboats,” Master’s thesis, Instituto Superior de Engenharia do Porto, 2019. Acedido em maio de 2020. [citado na p. 88]
- [92] C. Tretow, “Design of a free-rotating wing sail for an autonomous sailboat,” Master’s thesis, KTH - Royal Institute Of Technology, School Of Engineering Sciences, 2017. Acedido em setembro de 2020. [citado na p. 89]
- [93] ROS, “urdf/ xml/ link.” <http://wiki.ros.org/urdf/XML/link>, 2021. Acedido em agosto de 2021. [citado na p. 94]
- [94] ROS, “urdf/ xml/ joint.” <http://wiki.ros.org/urdf/XML/joint>, 2018. Acedido em agosto de 2021. [citado na p. 96]
- [95] SDFFormat, “urdf.” <http://sdformat.org/>, 2020. Acedido em setembro de 2021. [citado na p. 98]
- [96] ROS, “urdf.” <http://wiki.ros.org/urdf>, 2019. Acedido em setembro de 2021. [citado na p. 98]
- [97] ROS, “xacro.” <http://wiki.ros.org/xacro>, 2021. Acedido em setembro de 2021. [citado na p. 99]
- [98] A. Visioli, “Practical PID Control,” in *Springer Science | & Business Media*, 2006. Acedido em julho de 2021. [citado na p. 101]

Anexo A

Nomenclatura e Termos Náuticos

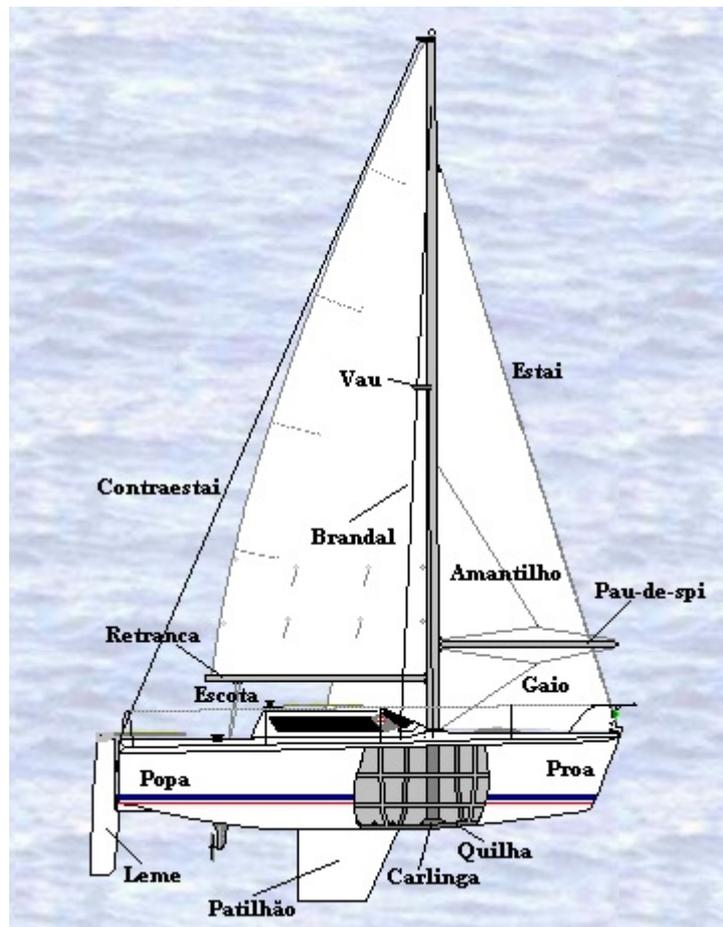


Figura A.1: Partes principais de um veleiro [49].

Tabela A.1: Descrição das partes principais de um veleiro.

Nome	Descrição
Casco	Concha externa do barco que contém todos os componentes internos do veleiro.
Cana do Leme	Posicionado na parte de trás do veleiro, ou popa, é utilizado para variar o ângulo do leme.
Leme	Componente que fica submersos na água. É responsável pelo direcionamento do barco.
Quilha/Patilhão	Localizada na parte de baixo do veleiro, permite que o barco não emborque, atuando como um peso de equilíbrio sob a água.
Mastro	Viga em que se prende o lado vertical da vela. Serve para fixar a vela principal ao veleiro.
Retranca	Viga em que se prende o lado horizontal da vela. Esse componente serve como braço de apoio para girar a vela e assim variar a incidência do vento na mesma. Desta forma é possível variar a velocidade do veleiro.
Vela Principal	Consiste na maior vela e a responsável por receber a maior parte do poder eólico necessário para mover o veleiro.
Burrajona	Vela secundária, menor que a principal, que permite uma potência adicional à vela principal.

Tabela A.2: Áreas de embarcação.

Nome	Descrição
Popa	Parte traseira do barco.
Proa	Parte dianteira do barco.
Estibordo	Corresponde ao lado direito.
Bombordo	Correspondente ao lado esquerdo.
Meia-Nau	Parte entre a proa e a popa.
Alhetas	Parte normal curva entre a meia-nau e a popa.
Través	Parte lateral da embarcação.
Amuras	Parte curva entre a meia-nau e a proa.
Convés	Patamar exterior principal.
Vante	Da meia-nau para a frente.
Ré	Da meia-nau para trás do navio.
Porão	Compartimento geralmente na proa, usado para guardar material da embarcação.
Genoa/Estai	Vela da proa.

Tabela A.3: Termos náuticos.

Nome	Descrição
Caçar	Puxar um cabo.
Folgar	Dar mais cabo.
Guinar/Virar de bordo	Virar a embarcação para bombordo ou estibordo.
Orçar	Aproximar a proa da linha do vento.
Arribar	Afastar a proa da linha do vento.
Fundiar	Largar a âncora.
Cambar	Rodar a linha de vento pela popa.
Bordo	Rodar a linha de vento pela proa.
Rumo	Trajeto real praticado pela embarcação.
Proa	Direção onde a proa aponta.
Aproar ao vento	Pôr a proa na linha do vento.
Barlavento	Sentido do vento, de onde vem.
Sotavento	Sentido do vento, para onde vai.

Anexo B

Ambiente ROS e Gazebo

Este anexo tem como finalidade servir como um guia sobre todas as etapas realizadas na configuração dos ambientes ROS e Gazebo.

B.1 ROS

Para instalar o ROS deve-se seguir os passos apresentados no *site* oficial (<https://www.ros.org/>). A versão do ROS selecionada para esta dissertação foi a versão Kinetic. As etapas de instalação são apresentadas em <http://wiki.ros.org/kinetic/Installation/Ubuntu>.

1. Ctrl + Alt + T para abrir um terminal da linha de comandos.
2. O seguinte comando serve para adicionar os pacotes ROS ao seu computador: `sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu`
3. Utilize o seguinte comando para aceder às chaves do servidor: `sudo apt install curl`. Para instalar o comando curl execute `curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -`
4. Utilize o seguinte comando para atualizar o sistema: `sudo apt-get update`
5. Utilize o seguinte comando para instalar a versão do ROS completa (e recomendada, ROS, rqt, rviz, bibliotecas genéricas de robôs, simuladores 2D e 3D e navegação e percepção 2D e 3D) `sudo apt-get install ros-kinetic-desktop-full`
6. Utilize o seguinte comando para instalar várias ferramentas e dependências para compilar e executar o ROS: `sudo apt install python-rosdep`

```
python-rosinstall python-rosinstall-generator python-wstool build-essential
```

7. Para instalar facilmente as dependências do sistema, execute os seguintes comandos: `sudo apt install python-rosdep -> sudo rosdep init -> rosdep update`
8. Para ter acesso aos comandos ROS execute o comando `source /opt/ros/kinetic/setup.bash`. Este comando precisa ser executado sempre que um novo terminal for aberto.
9. Para criar o espaço de trabalho ROS é necessário fazer o seguinte: `mkdir -p /catkin_ws/src -> cd catkin_ws -> catkin_make`
10. Com o intuito de automatizar a etapa 8 é necessário editar o ficheiro de configuração da *shell* com o comando `gedit .bashrc` e adicionar uma nova linha contendo `source yourlocation/catkin_ws/devel/setup.bash`, como demonstrado na Listagem B.1. A configuração do ambiente ROS está concluída.

Listagem B.1: Ficheiro `.bashrc`.

```

1 #enable programmable completion featura (you don't need to enable
2 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
3 #sources /etc/bash.bashrc).
4 if ! shopt -oq posix; then
5     if [ -f /usr/share/bash-completion/bash_completion ]; then
6         . /usr/share/bash-completion/bash_completion
7     elif [ -f /etc/bash_completion ]; then
8         . /etc/bash_completion
9     fi
10 fi
11
12 source /opt/ros/kinetic/setup.bash
13 source ~/catkin_ws/devel/setup.bash
14
15 export GAZEBO_PREFIX=$HOME/catkin_ws/install
16 export GAZEBO MODEL_PATH=${GAZEBO_PREFIX}/share/gazebo-7.16.1/models:${GAZEBO
    MODEL_PATH}

```

B.2 Gazebo

Com o ROS instalado é necessário instalar o Gazebo. As seguintes etapas são apresentadas em http://gazebosim.org/tutorials?cat=guided_b&tut=guided_b1

1. Fazer *download* do Gazebo em <http://gazebosim.org/> e guardar na pasta Downloads.

2. Para abrir o terminal de comandos execute `Ctrl + Alt + T`.
3. Na linha de comandos execute o comando `chmod +x /Downloads/gazebo.sh` para transformar o ficheiro transferido num ficheiro executável.
4. Execute o comando `gnome-terminal -working-directory= -e "./Downloads/gazebo-sh"` para iniciar a instalação do Gazebo.
5. Aparecerá uma nova janela para inserir a senha do sistema.
6. Após a inserção da senha, o Gazebo deve ser instalado no sistema.

Anexo C

Instalação de Plugin

Antes de criar, por exemplo, o *plugin* do anemómetro, os ficheiros de desenvolvimento do Gazebo devem ser instalados. O comando é o seguinte:

1. `sudo apt-get install libgazebo8-dev.`

Para criar um *plugin* seguem-se os passos apresentados em http://gazebo.org/tutorials/?tut=plugins_hello_world:

1. Primeiro, deve-se criar um novo directório: `mkdir catkin_ws/src/usv_sim_lsa/usv_sensor -> cd catkin_ws/src/usv_sim_lsa/usv_sensor.`
2. Criar um ficheiro `.cpp` para o *plugin*: `gedit anemometer.cpp` e um ficheiro `.h`: `gedit anemometer.h` para escrever o código do *plugin*.
3. Para compilar o *plugin*, deve-se criar a `CMakeLists.txt`: `gedit CMakeLists.txt.`
4. Criar o ficheiro `build`: `mkdir build -> cd build.`
5. Compilar o *plugin*: `cmake ../ -> make.`

A compilação do *plugin* resultará numa biblioteca partilhada, `~/catkin_ws/src/usv_sim/usv_sensor/build/libanemometer.so`, que pode ser inserida numa simulação do Gazebo. De forma a utilizar a biblioteca na pasta do espaço de trabalho ROS, deve-se copiar e colar este ficheiro `.so` para o directório `/devel/lib` dentro do espaço de trabalho ROS.

