



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO



Arquitetura de Software para Barcos Robóticos

Einstein Gomes dos Santos

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves

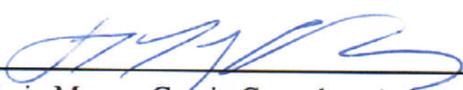
Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

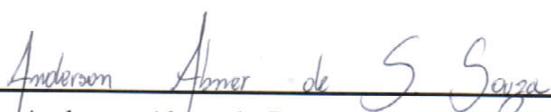
Natal, RN, janeiro de 2014

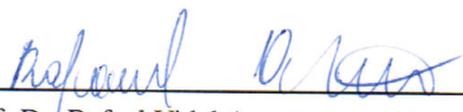
Arquitetura de software para Barcos Robóticos

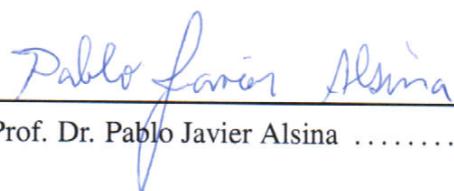
Einstein Gomes dos Santos

Dissertação de Mestrado aprovada em 23 de janeiro de 2014 pela banca examinadora composta pelos seguintes membros:


Prof. Dr. Luiz Marcos Garcia Gonçalves (orientador) DCA/UFRN


Prof. Dr. Anderson Abner de Santana Souza UERN


Prof. Dr. Rafael Vidal Aroca IMD/UFRN


Prof. Dr. Pablo Javier Alsina DCA/UFRN

*“Ĉiu efiko havas kaŭzon. Ĉiu inteligenta efiko havas
intelligentan kaŭzon. La povo de la inteligenta kaŭzo
devenas de la grandeco de la efiko.”*

Allan Kardec

Agradecimentos

A Deus sempre e a todos os Irmãos que laboram sob sua inspiração e que me amparam.

Aos meus queridos pais Eloilson Arcanjo e Maria Irene sou eternamente grato por tudo que fizeram e fazem por mim e pelos meus irmãos.

Ao meu orientador, o professor Luiz Marcos, sou grato pela orientação e ajuda.

Ao professor Paulo Motta por me ter acolhido em um primeiro momento na pós-graduação.

A todos os colegas de pesquisa do projeto N-Boat que de alguma forma me ajudaram.

Às minhas irmãs Elizabeth Cristina e Christiane Gomes pelas sugestões dadas ao longo da elaboração deste trabalho.

À minha esposa Hannah Mendes e à minha família pelo apoio durante esta jornada.

À CAPES, pelo apoio financeiro.

Resumo

Propomos neste trabalho uma arquitetura de *software* para barcos robóticos destinados a atuarem em ambientes aquáticos diversos, de forma totalmente autônoma, realizando telemetria com uma estação-base e desta recebendo missões a serem realizadas. Tal proposta visa aplicar-se dentro do projeto N-Boat do laboratório Natalnet-DCA, que tem como objetivo principal capacitar um veleiro a navegar autonomamente. Os componentes constituintes dessa arquitetura são os módulos de memória, estratégia, comunicação, sensoriamento, atuação, energia, segurança e supervisão, formando estes os sistemas do barco e da estação-base. Para sua validação foi desenvolvido um simulador implementado na linguagem C e utilizando recursos da API gráfica OpenGL, cujos principais resultados foram obtidos na implementação dos módulos de memória, de atuação e de estratégia, mais especificamente no compartilhamento de dados, no controle das velas e do leme e no planejamento de rotas curtas baseado em um algoritmo de navegação, respectivamente. Os resultados dos experimentos realizados, mostrados no presente trabalho, indicam a viabilidade da utilização real da arquitetura de *software* desenvolvida e sua aplicação na área da robótica móvel autônoma.

Palavras-chave: Sistemas Robóticos Autônomos, Veleiros Autônomos, Arquitetura de *Software*.

Abstract

We propose in this work a software architecture for robotic boats intended to act in diverse aquatic environments, fully autonomously, performing telemetry to a base station and getting this mission to be accomplished. This proposal aims to apply within the project N-Boat Lab NatalNet – DCA, which aims to empower a sailboat navigating autonomously. The constituent components of this architecture are the memory modules, strategy, communication, sensing, actuation, energy, security and surveillance, making these systems the boat and base station. To validate the simulator was developed in C language and implemented using the graphics API OpenGL resources, whose main results were obtained in the implementation of memory, performance and strategy modules, more specifically data sharing, control of sails and rudder and planning short routes based on an algorithm for navigation, respectively. The experimental results, shown in this study indicate the feasibility of the actual use of the software architecture developed and their application in the area of autonomous mobile robotics.

Keywords: Autonomous Robotic Systems, Autonomous Sailboats, Software Architecture.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Símbolos e Abreviaturas	vii
1 Introdução	1
1.1 O problema	1
1.2 Contribuições	2
1.3 Organização do texto	2
2 Embasamento Teórico	5
2.1 Navegação a vela	5
2.1.1 Partes de um veleiro	5
2.1.2 Maneiras de se velejar em relação ao vento	7
2.1.3 Mareações	7
2.1.4 Mudanças de curso	9
2.2 Instrumentos náuticos	10
2.2.1 Instrumentos para medida de direções no mar	10
2.2.2 Instrumentos de medida de velocidade e distância percorrida	11
2.2.3 Instrumentos para medida de profundidade	12
3 Estado da Arte	15
3.1 ASV Roboat	15
3.2 FASt	17
3.3 Avalon	20
3.4 Dados técnicos dos veleiros autônomos estudados	23
3.5 Comparação entre projetos	26

4	O Problema e a Solução	27
5	Implementação	31
5.1	Simulador	31
5.2	Algoritmo de controle	32
5.3	Algoritmo de navegação	32
5.4	Algoritmo de controle das velas	34
5.5	Algoritmo de controle do leme	36
6	Experimentos e Resultados	39
7	Conclusões e Trabalhos Futuros	55
	Referências bibliográficas	57
A	Principais termos náuticos	59
B	Algoritmo de navegação modificado	61

Lista de Figuras

1.1	Nautimodelo RC Monsoon 900.	3
2.1	Partes básicas de um veleiro.	6
2.2	Maneiras de se velejar em relação ao vento. A) velejando a favor do vento; B) velejando em ângulo reto em relação ao vento; e C) velejando contra o vento.	8
2.3	Navegação contra o vento. A) área proibida para navegação em um curso reto; B) forças atuantes em um barco ao navegar contra o vento.	8
2.4	Mareações que um veleiro pode adotar.	9
2.5	Mudanças de curso que um veleiro pode executar.	10
2.6	Componentes do sistema de navegação DGPS em tempo real.	13
3.1	Diagrama da arquitetura de subsunção híbrida em quatro camadas do ASV Roboat.	16
3.2	Sistema de comunicação de três estágios do ASV Roboat. A) Primeiro estágio; B) Segundo estágio; C) Terceiro estágio.	18
3.3	Organização do <i>software</i> do Avalon.	21
3.4	Diagrama da arquitetura do <i>software</i> do Avalon.	22
3.5	<i>Waypoints</i> calculados pelo <i>navigator</i>	22
3.6	ASV Roboat.	23
3.7	FASt.	24
3.8	Avalon.	25
4.1	Diagrama da arquitetura de <i>software</i> da plataforma.	28
4.2	Diagrama da arquitetura de <i>software</i> da estação-base.	30
5.1	Gráfico do polinômio calculado, utilizado no controle das velas, que fornece a posição destas a partir da posição do barco em relação ao vento.	35
6.1	Experimento 1a: navegação em linha reta (1° quadrante).	42
6.2	Experimento 1b: navegação contra o vento (1° quadrante).	43

6.3	Experimento 1c: navegação a favor do vento (1° quadrante).	44
6.4	Experimento 2a: navegação em linha reta (2° quadrante).	45
6.5	Experimento 2b: navegação contra o vento (2° quadrante).	46
6.6	Experimento 2c: navegação a favor do vento (2° quadrante).	47
6.7	Experimento 3a: navegação em linha reta (3° quadrante).	48
6.8	Experimento 3b: navegação contra o vento (3° quadrante).	49
6.9	Experimento 3c: navegação a favor do vento (3° quadrante).	50
6.10	Experimento 4a: navegação em linha reta (4° quadrante).	51
6.11	Experimento 4b: navegação contra o vento (4° quadrante).	52
6.12	Experimento 4c: navegação a favor do vento (4° quadrante).	53

Lista de Tabelas

3.1	Dados do ASV Roboat.	23
3.2	Dados do FASt.	24
3.3	Dados do Avalon.	25
3.4	Comparação entre projetos de veleiros autônomos.	26
4.1	Estimativa de componentes do veleiro N-Boat.	27
4.2	Disposição dos componentes do N-Boat na arquitetura da plataforma. . .	29
6.1	Estudo comparativo das distâncias entre o ponto P_0 e os pontos $P_{1,2,3,4}$. . .	40
6.2	Resultados referentes ao experimento 1a.	42
6.3	Resultados referentes ao experimento 1b.	43
6.4	Resultados referentes ao experimento 1c.	44
6.5	Resultados referentes ao experimento 2a.	45
6.6	Resultados referentes ao experimento 2b.	46
6.7	Resultados referentes ao experimento 2c.	47
6.8	Resultados referentes ao experimento 3a.	48
6.9	Resultados referentes ao experimento 3b.	49
6.10	Resultados referentes ao experimento 3c.	50
6.11	Resultados referentes ao experimento 4a.	51
6.12	Resultados referentes ao experimento 4b.	52
6.13	Resultados referentes ao experimento 4c.	53

Lista de Símbolos e Abreviaturas

3G	3rd Generation mobile telecommunications
API	Application Programming Interface
ASV	Autonomous Surface Vehicle
DGPS	Differential Global Positioning System
DoD	Department of Defense
ENSTA	École Nationale Supérieure de Techniques Avancées
ETH	Swiss Federal Institute of Technology Zurich
FASt	FEUP Autonomous Sailboat
FEUP	Faculdade de Engenharia da Universidade do Porto
GPRS	General Packet Radio Service
GPS	Global Positioning System
IFREMER	Institut Français de Recherche pour l'Exploitation de la Mer
INNOC	Austrian Society for Innovative Computer Sciences
IRSC	International Robotic Sailing Conference
ISAE	Institut Supérieur de l'Aéronautique et de l'Espace
LAN	Local Area Network
OEM	Original Equipment Manufacturer
PPS	Precise Positioning Service
SPS	Standard Positioning Service

USCGA United States Coast Guard Academy

USNA United States Naval Academy

VPN Virtual Private Network

Wp Watt pico

WRSC World Robotic Sailing Championship

Capítulo 1

Introdução

Cerca de dois terços da superfície da Terra é coberta por água, sendo esta formada pelos lagos, lagoas, rios, geleiras, calotas polares e o mar. Desses, o que possui maior predominância é o mar. A sua influência na vida no planeta é enorme abrangendo desde a regulação do clima até o fornecimento de alimentos, recursos minerais e energéticos. Apesar da ostensiva extração realizada de seus recursos naturais, muito pouco se sabe sobre sua potencial riqueza, mostrando-se, desta forma, incomensurável.

Essa situação é explicada, em parte, pelo ainda escasso conhecimento que possuímos sobre sua natureza e seus mecanismos de ação. Aliada a esse cenário, está a crescente degradação do meio ambiente marinho, tendo como autor o homem que, mediante suas atividades, o polui de diversas maneiras. Disso, torna-se extremamente necessário e urgente o desenvolvimento de sistemas de monitoramento que levantem dados que, após serem analisados e estudados, nos levem a uma maior compreensão da realidade do estado do mar e a elaborar mecanismos que nos proporcione meios de protegê-lo e explorá-lo de forma mais sustentável.

Devido as grandes dimensões da área a monitorar, surge a necessidade de desenvolvimento de sistemas de monitoramento que ofereçam recursos necessários para a realização dessa tarefa. Nesse contexto, a utilização de sistemas automáticos de monitoramento é de grande serventia, auxiliando de forma complementar os sistemas de sensoriamento remoto, já existentes.

1.1 O problema

O desenvolvimento de uma arquitetura de *software* para uma plataforma robótica aquática autônoma (veleiro robótico) tem por base o gerenciamento de todos os módulos que compõe o seu sistema, de forma a capacitá-la a navegar tomando decisões autonomamente. Quando em missão, a plataforma deve determinar o melhor rumo a ser tomado,

tendo como parâmetros o ponto alvo fornecido pela estação-base e os dados coletados pelos seus sensores de bordo, e realizar telemetria. Para seguir o rumo traçado é preciso escolher a melhor estratégia de navegação, realizando as manobras necessárias atuando nas velas e no leme, e se resguardando de situações de risco como mau tempo e colisão de obstáculos.

Diante do que foi dito, uma arquitetura de *software* básica voltada para uma plataforma robótica aquática autônoma precisa gerenciar os seguintes módulos: módulo de sensoriamento, destinado a captar todos os dados relevantes ao processamento do sistema; módulo de comunicação, responsável pela troca de dados entre a estação-base e a plataforma, e vice-versa; módulo de atuação, destinado ao controle das velas e do leme; módulo de segurança, responsável pela detecção de mau tempo e colisão de obstáculos; e o módulo de estratégia, destinado a tomadas de decisão referentes a uma navegação segura e otimizada, tendo como base informações provindas de outros módulos.

1.2 Contribuições

A contribuição principal deste trabalho é uma arquitetura de *software* destinada a plataformas robóticas aquáticas autônomas constituída por uma configuração diferenciada dos módulos gerenciáveis que a compõe. Tal arquitetura foi inspirada no veleiro autônomo Avalon, que utiliza no gerenciamento dos módulos de seu sistema uma memória compartilhada. A justificativa para a sua adoção foi a facilidade que proporciona no tratamento da complexidade do gerenciamento dos módulos do sistema.

Além da arquitetura de *software* proposta, este trabalho também apresenta algoritmos destinados à geração de rotas curtas de navegação e ao controle das velas e do leme.

Este trabalho foi implementado, de forma simplificada, em um simulador baseado na biblioteca gráfica OpenGL. No futuro, este trabalho fará parte do sistema do protótipo, o nautimodelo rádio-controlado mostrado na Figura 1.1 e, mais adiante, do sistema do veleiro robótico autônomo N-Boat a ser adquirido.

1.3 Organização do texto

Este documento está dividido em sete capítulos.

- O capítulo 1 apresentou o contexto no qual está inserido este trabalho, bem como suas contribuições e aplicações;

- O capítulo 2 mostra o embasamento teórico sobre navegação a vela e os principais instrumentos náuticos;
- O capítulo 3 apresenta o estudo realizado sobre a arquitetura de controle dos principais veleiros autônomos pesquisados;
- O capítulo 4 trata do problema da implementação da arquitetura de controle e da solução encontrada;
- O capítulo 5 discorre sobre os algoritmos que foram desenvolvidos para serem implementados no simulador. A saber: o algoritmo de controle geral, de navegação, de controle das velas e de controle do leme;
- O capítulo 6 apresenta os experimentos realizados e os resultados obtidos;
- Por fim, o capítulo 7 apresenta as conclusões chegadas e as novas perspectivas sobre este trabalho.



Figura 1.1: Nautimodelo RC Monsoon 900.
Fonte: Hobbycontrol (2013).

Capítulo 2

Embasamento Teórico

2.1 Navegação a vela

Velejar é uma das mais antigas práticas que o homem desenvolveu para ajudá-lo na sua locomoção e no transporte de carga, a longa distância, utilizando os recursos proporcionados pelo meio aquático e pelo vento. O veleiro é uma máquina utilizada para velejar, que utiliza a força do vento captada pelas velas para movimentá-lo para a frente, através da água.

2.1.1 Partes de um veleiro

Existem diversos tipos de veleiros, que variam tanto em relação ao formato e tamanho quanto a finalidade. Na Figura 2.1 encontram-se descritas as partes básicas de um veleiro, cujas definições seguem abaixo. O Apêndice A apresenta outros termos náuticos importantes.

Mastro: Haste perpendicular ao barco usada para sustentar as velas.

Vela mestra: Ou principal, ou ainda grande, é a maior vela do barco.

Buja: Pequena vela que se localiza na proa do barco.

Proa: Extremidade dianteira do barco.

Popa: Extremidade traseira do barco.

Bombordo: Lado esquerdo do barco.

Boreste: Ou estibordo, é o lado direito do barco.

Retranca: Haste unida à parte inferior da vela.

Escota: Cabo ligado à retranca, cuja função é regular a abertura ou o fechamento das velas.

Leme: Peça laminar utilizada para modificar a direção do barco.

Quilha: Peça em forma de barbatana fixada na parte de baixo do casco do veleiro, cuja função é impedir o abatimento e auxiliar na estabilidade.

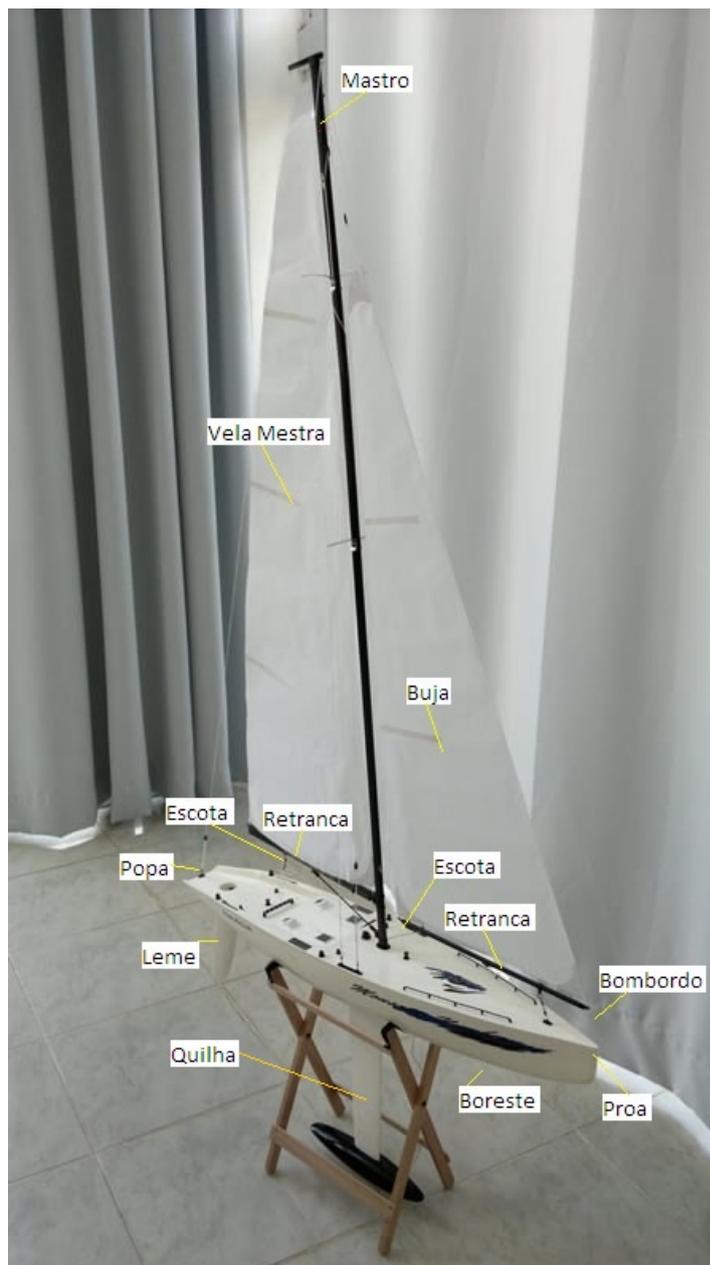


Figura 2.1: Partes básicas de um veleiro.
Fonte: Hobbycontrol (2013).

2.1.2 Maneiras de se velejar em relação ao vento

Segundo Dent (1978), dependendo do ângulo formado pelo curso que o veleiro segue e a direção que o vento sopra, o veleiro poderá velejar ou a favor do vento, ou em ângulo reto em relação ao vento ou contra o vento.

No caso de estar velejando a favor do vento, as velas formam em relação a cada lado do barco um ângulo reto para captar a maior quantidade possível de vento. O vento empurra as velas ao encontrá-las, fazendo com que o barco se mova. A velocidade obtida pelo barco é menor do que a do vento devido às perdas ocasionadas pela turbulência gerada pelo movimento do vento em contorna as velas. Essa situação é ilustrada, de forma simplificada, na Figura 2.2A.

Quando o veleiro está velejando em ângulo reto em relação ao vento, esquema representado na Figura 2.2B, o vento colide com as velas, que formam um determinado ângulo com um dos lados do barco, e flui paralelamente a estas. Com isso, não ocorre um bloqueio em seu fluxo, como é visto na Figura 2.2A, e sim um desvio deste. A quilha influencia nessa forma de velejar impedindo que o barco se desloque para os lados. Nessa forma de velejar, certos barcos conseguem desenvolver maior velocidade do que o vento.

Na situação de velejar contra o vento, as velas encontram-se posicionadas formando um ângulo menor ainda, em relação a um dos lados do barco, do que quando se veleja em ângulo reto em relação ao vento, fazendo com que o vento se desvie mais ainda. O barco é empurrado para frente, porém com menor velocidade. A quilha aqui é mais solicitada ainda do que na forma de velejar anterior. O esquema é descrito na Figura 2.2C.

Na verdade, nenhum barco pode velejar diretamente contra o vento. Há uma região situada a aproximadamente 45 graus de ambos os lados da direção de onde o vento sopra, conhecida como área proibida ou morta, na qual não é possível atingir determinado ponto em linha reta. Para que esse ponto possa ser atingido, o barco precisa velejar em zigue-zague formando com a direção do vento um ângulo de 45 graus. Essa maneira de velejar recebe o nome de bordejar e cada zigue e cada zague recebe o nome de perna ou bordo. A área proibida é ilustrada na Figura 2.3A. Na Figura 2.3B são mostradas as forças atuantes em um veleiro quando ele navega contra o vento. O vetor 2 representada a força total, o vetor 3 a força lateral e o vetor 1 a força que impulsiona o barco para frente.

2.1.3 Mareações

Mareações é o nome que se dá as várias posições que as velas assumem em relação ao vento, e possuem diferentes denominações, as quais se encontram apresentadas na Figura 2.4.

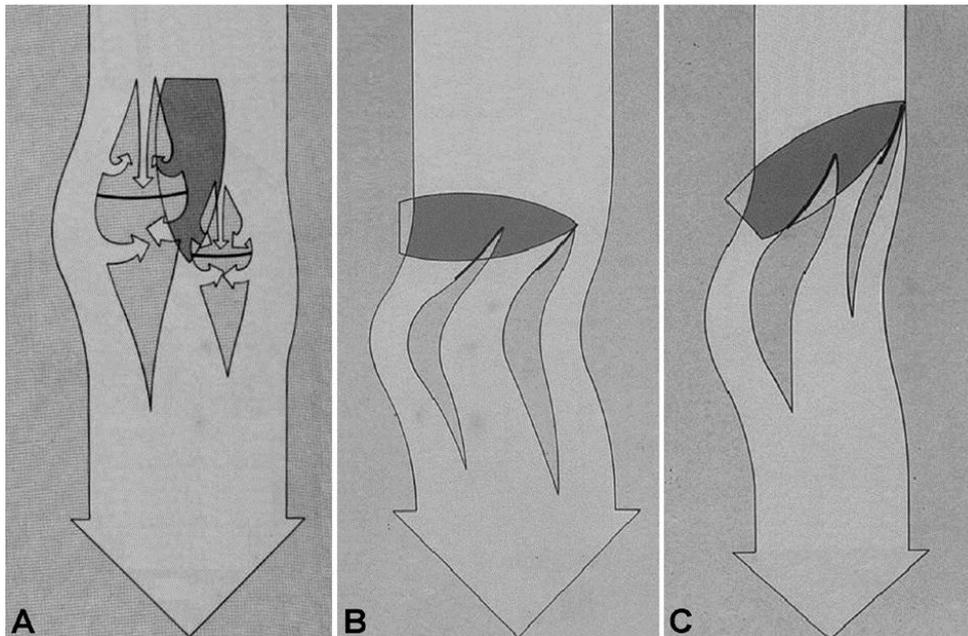


Figura 2.2: Maneiras de se velejar em relação ao vento. A) velejando a favor do vento; B) velejando em ângulo reto em relação ao vento; e C) velejando contra o vento.

Fonte: Dent (1978).

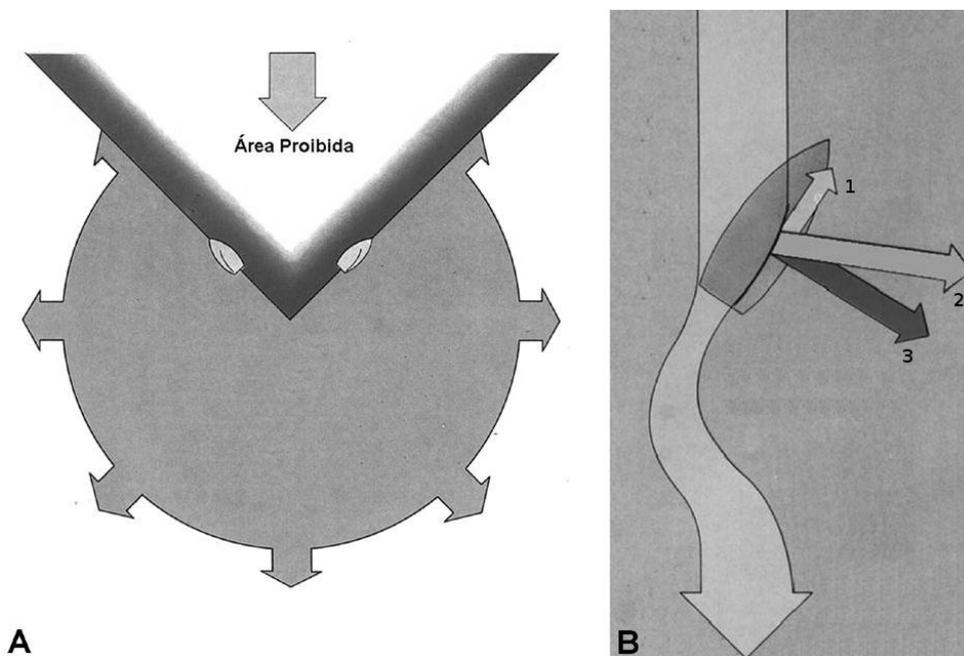


Figura 2.3: Navegação contra o vento. A) área proibida para navegação em um curso reto; B) forças atuantes em um barco ao navegar contra o vento.

Fonte: Dent (1978).

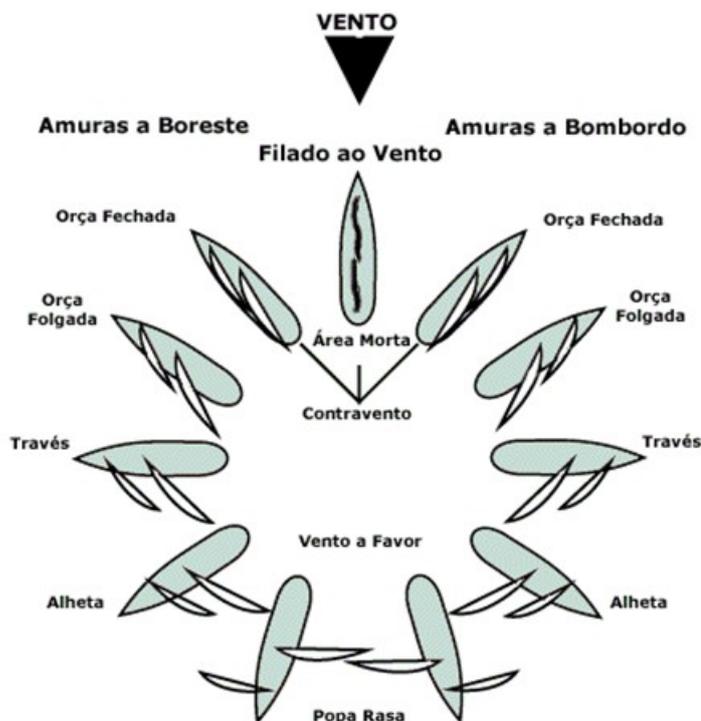


Figura 2.4: Mareações que um veleiro pode adotar.
Fonte: Francisco (2013).

2.1.4 Mudanças de curso

Existem quatro tipos básicos de mudança de curso que um veleiro pode executar. Quando o barco aproxima a proa da direção do vento, diz-se que o barco está orçando; quando ao contrário, ele afasta a proa da direção do vento, ele está arribando; quanto mais se arriba, mais são folgadas as escotas das velas, e quanto mais se orça mais são caçadas, ou seja, puxadas as escotas das velas.

As velas cruzam o barco de um lado, ou bordo, para o outro a partir de duas manobras conhecidas por: virar em roda e virar por davante. Virar em roda ou jibar ocorre quando as velas cruzam de um bordo ao outro e o vento incide diretamente na popa do barco. Enquanto que, virar por davante ou bordejar ou ainda cambar ocorre quando as velas cruzam de um bordo ao outro e o vento incide diretamente na proa da embarcação. A Figura 2.5 apresenta um diagrama no qual podem ser vistos os quatro tipos básicos de mudanças de curso discutidos.

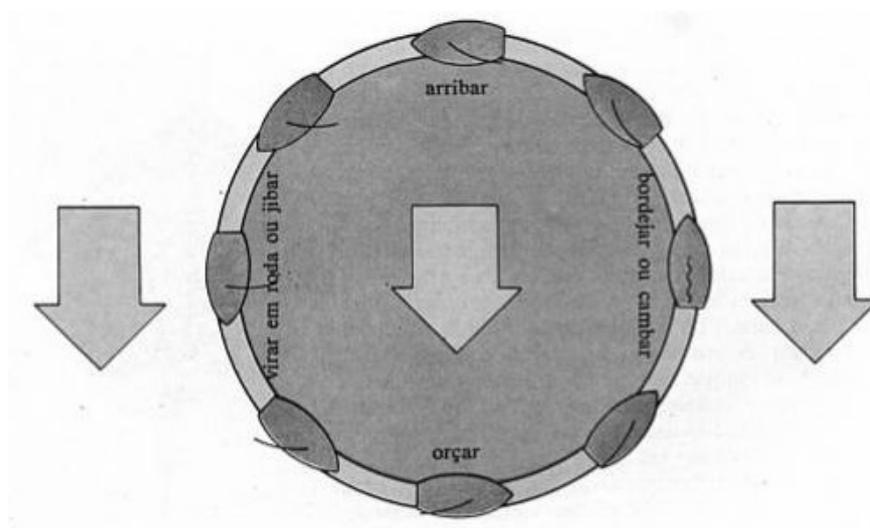


Figura 2.5: Mudanças de curso que um veleiro pode executar.
Fonte: Dent (1978).

2.2 Instrumentos náuticos

Seguindo a classificação adotada por Miguens (2000), serão apresentados, de forma não aprofundada, alguns dos instrumentos náuticos utilizados em embarcações.

2.2.1 Instrumentos para medida de direções no mar

Agulha eletrônica

É uma bússola cujo funcionamento se baseia na medida do campo magnético terrestre, apresentando o rumo de forma geralmente digital, não existindo partes móveis em sua constituição, possuindo a capacidade de ser utilizada em latitudes mais elevadas que as bússolas convencionais.

Agulha Giroscópica

É uma bússola precisa e sofisticada que utiliza um giroscópio eletrônico e mancais magnéticos.

Acessórios das agulhas giroscópicas:

Piloto automático: É um aparelho que controla automaticamente o rumo de um navio, possibilitando este manter a navegação em um determinado rumo sem a intervenção do timoneiro.

Registrador de rumos: É um instrumento que possui a capacidade de registrar os rumos navegados por uma embarcação em função do tempo.

2.2.2 Instrumentos de medida de velocidade e distância percorrida

Odômetros

Conforme Miguens (2000), os odômetros podem ser classificados em:

- odômetro de superfície;
- odômetro de fundo; e
- odômetro Doppler.

Os odômetros de superfície e de fundo medem a velocidade do navio em relação à massa de água circundante, sendo que a distância percorrida é obtida pela integração da velocidade em relação ao tempo por meios elétricos ou mecânicos. O odômetro Doppler tem a capacidade de medir a velocidade do navio em relação ao fundo.

O odômetro de superfície é constituído por uma hélice, um volante, uma linha de reboque e um contador. Quando a embarcação se desloca a hélice gira e esse movimento é transmitido mediante a linha de reboque e o volante ao contador que mostra o total da distância navegada desde o momento em que o instrumento foi inserido na água.

O odômetro de fundo tipo pressão utiliza um tubo de Pitot que mede a pressão dinâmica, isto é, a pressão ocasionada pelo deslocamento da embarcação através da água. A partir da pressão dinâmica é obtida a velocidade que, por sua vez, é convertida em distância navegada.

O odômetro de fundo tipo eletromagnético baseia-se no princípio da produção de uma força eletromotriz (FEM) a partir da variação de um campo magnético. A FEM induzida gerada pelo deslocamento da embarcação é proporcional à velocidade em relação à água, que é convertida em distância navegada.

O odômetro Doppler é composto por um transdutor de emissão e um de recepção. Eles são utilizados para emitir e receber, respectivamente, um sinal de frequência ultrassônica, cuja reflexão do sinal acontece no fundo do mar. Quando a embarcação está em movimento, a frequência do sinal emitido é diferente da frequência do sinal recebido. A diferença de frequências é diretamente proporcional à velocidade da embarcação que, após ser integrada, fornece a distância navegada.

Velocímetros

São instrumentos que medem a velocidade da embarcação em relação à água e são geralmente atuados por uma haste submersa que é pressionada para trás quando a embarcação se encontra em movimento.

No velocímetro hidráulico, a inclinação da haste é transmitida ao êmbolo de um cilindro que comprime um líquido associado ao indicador do velocímetro.

No velocímetro de hélice, na haste se localiza uma hélice que gira devido ao movimento da embarcação e alimenta um gerador de corrente alternada. A velocidade da embarcação é proporcional à frequência da corrente gerada. Existe também o velocímetro que conta eletricamente as rotações da hélice para fornecer a velocidade.

GPS

Segundo Miguens (2000), o Sistema de Posicionamento Global, mais conhecido pela abreviatura GPS, é formado por três partes principais: satélites; sistema de monitoramento e controle; e aparelhos receptores GPS e equipamentos de apoio. As três partes operam em conjunto, fornecendo latitude, longitude, altitude, rumo, velocidade e hora, com alta precisão.

O GPS disponibiliza dois serviços, o Serviço de Posicionamento Preciso (PPS) e o Serviço Padrão de Posicionamento (SPS). O primeiro é utilizado apenas pelas forças armadas dos EUA e seus aliados da OTAN e o segundo por qualquer usuário, sendo seu nível de precisão degradado intencionalmente pelo Departamento de Defesa do EUA (DoD).

Para aumentar a precisão do Serviço Padrão de Posicionamento, foi desenvolvida a técnica diferencial aplicada ao GPS. Essa técnica corrige a degradação intencional, influências atmosféricas, erros de sincronização dos relógios dos satélites e irregularidades em suas órbitas. Na Figura 2.6 são apresentados os componentes do sistema de navegação DGPS.

2.2.3 Instrumentos para medida de profundidade

Ecobatímetros

Também conhecido como sonda sonora, é um dispositivo que utiliza feixes de ondas sonoras ou ultrassônicas para medir a profundidade do fundo do mar. Conhecendo a velocidade do som na água, basta medir o tempo decorrido entre a emissão do sinal e a recepção do eco refletido do fundo para se obter a profundidade.

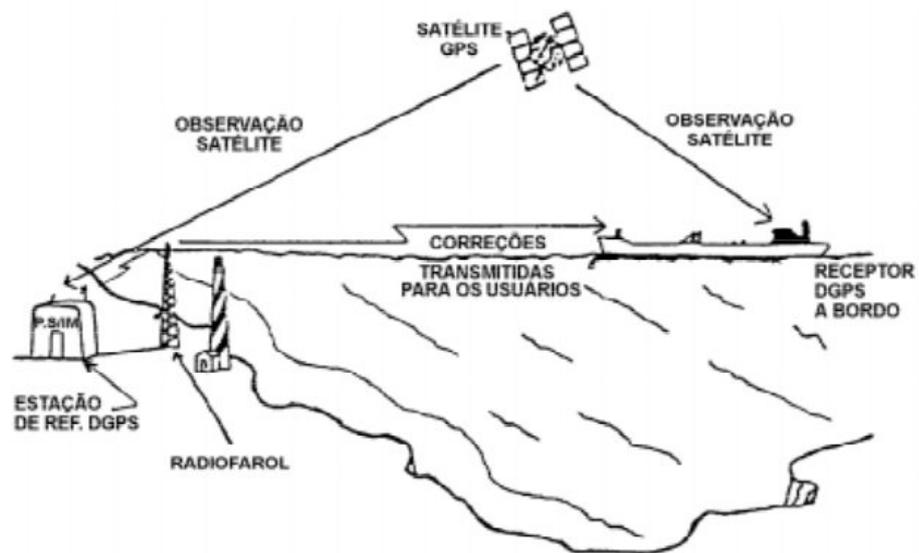


Figura 2.6: Componentes do sistema de navegação DGPS em tempo real.
Fonte: Miguens (2000).

Capítulo 3

Estado da Arte

Vários projetos de veleiros autônomos foram desenvolvidos e, como exemplos, podemos citar os seguintes:

- *Däumling, Pi-mal-Daumen* [Stelzer 2012] (Universidade de Lübeck / Alemanha);
- *MOOP, ARC, AROO, Beagle-B, Pinta* [Sauzé 2005] (Universidade de Aberystwyth / Reino Unido);
- *Breizh Spirit* [Sliwka et al. 2009] (ENSTA / França);
- *Vaimos* [Jaulin & Bars 2013] (IFREMER – ENSTA / França);
- *Black Adder, North Star, Gaelforce, Gaelforce II* [QMAST 2013] (Universidade de Queen / Canadá);
- *First Time, Gill the Boat, Luce Canon* [Miller et al. 2009] (USNA / EUA);
- *Intuition* [Stelzer 2012] (USCGA / EUA);
- *IBoat* [Briere 2008] (ISAE / França);
- *GreenSails [INCT-SEC 2013] (INC-SEC / Brasil)*
- *FASt* [Alves & Cruz 2008] (FEUP / Portugal);
- *Avalon* [Erckens et al. 2010] (ETH / Suíça);
- *Roboat I, ASV Roboat* [Stelzer 2012] (INNOC / Áustria).

Dos projetos de veleiros autônomos citados, três se destacam pelos seus desempenhos no campo prático e pela quantidade de material de pesquisa publicada. São eles: o *ASV Roboat*; o *FASt*; e o *Avalon*.

3.1 ASV Roboat

O veleiro autônomo ASV Roboat foi construído, como segundo protótipo pela equipe da INNOC, para participar do Desafio Microtransat [Microtransat 2013] de 2007 ocorrido no mar da Irlanda [Stelzer 2012].

Seu *software* de controle é executado numa distribuição do sistema operacional Linux no computador de bordo que está conectado aos sensores e aos atuadores. O sistema de sensoriamento é parcialmente redundante e mede as seguintes grandezas: velocidade e direção do vento, direção do barco, velocidade angular, adernamento, posição do barco, velocidade do barco, profundidade da água, temperatura da água, temperatura do ar, umidade do ar e pressão atmosférica. O veleiro é equipado com atuadores para as velas, leme, luz de navegação, buzina de nevoeiro e bomba de porão.

O ASV Roboat possui duas fontes de energia: a principal, que são os painéis solares que podem fornecer até 285 Wp, que abastecem um banco de baterias de 1,92 kWh; e a de emergência que é constituída de uma célula combustível de metanol que fornece 65 W.

A arquitetura do sistema é composta por duas partes principais: a arquitetura de subsunção híbrida em quatro camadas (Figura 3.1) e o sistema de comunicação em três estágios (Figura 3.2).

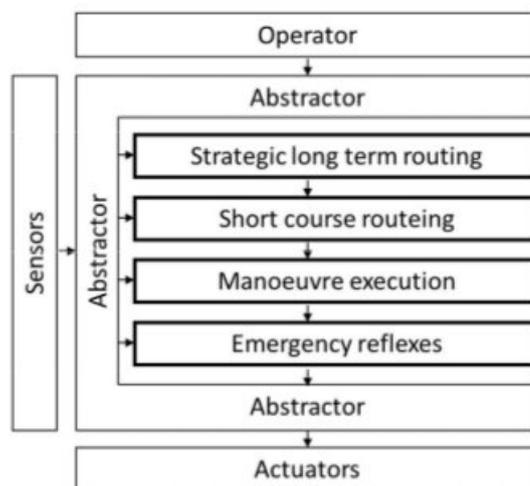


Figura 3.1: Diagrama da arquitetura de subsunção híbrida em quatro camadas do ASV Roboat.

Fonte: Stelzer & Jafarmadar (2007).

A arquitetura de subsunção híbrida combina as abordagens deliberativa e a reativa e é constituída por quatro camadas: roteamento estratégico de longo prazo, roteamento de curso curto, execução de manobra e reflexos de emergência. Elas são executadas de forma assíncrona e em paralelo. Cada camada acessa os dados oriundos dos sensores mediante o abstrator, que é um programa que é executado no computador de bordo do veleiro. Esse programa é responsável por processar os dados brutos entregues pelos sensores, transformando-os em dados inteligíveis para as camadas.

Roteamento estratégico de longo prazo

É executado o algoritmo de roteamento que define uma rota ótima tendo como base determinados parâmetros, como, por exemplo, as condições do tempo e a topologia do mar. A rota é dividida em várias sub-rotas e as coordenadas dos pontos de união dessas sub-rotas são passadas sequencialmente para a camada de roteamento de curso curto.

Roteamento de curso curto

Determina um caminho ótimo para atingir o ponto alvo dado pela camada de roteamento estratégico de longo prazo. São levados em consideração para a escolha do melhor caminho as condições do vento e o tipo de maré a ser seguido.

Execução de manobra

Recebe da camada de roteamento de curso curto uma direção desejada que o veleiro deve seguir. Caso ocorra desvio do curso, o sistema pertencente a essa camada ajusta a posição do leme para colocar o barco novamente no curso desejado. Um segundo sistema é executado em paralelo para garantir fluxo de ar nas velas para que possa haver propulsão. Esta camada tem de executar cambagem ou jibe caso seja preciso ao longo da navegação.

Reflexos de emergência

Recebe da camada de execução de manobra as posições desejadas do leme e das velas. Essas posições são repassadas inalteradas aos atuadores quando não existe uma ameaça à segurança do barco. Caso exista, as ações solicitadas são ignoradas.

Em relação ao sistema de comunicação pertencente ao ASV Roboat, este é constituído por três estágios que oferece os recursos necessários para que haja a troca de dados entre o veleiro, o supervisor e o controlador remoto. Ocorre chaveamento dinâmico automático entre os canais de comunicação disponíveis no veleiro. Os estágios são os seguintes:

Primeiro estágio: *wireless* LAN (Figura 3.2A);

Segundo estágio: serviço de dados 3G de provedor de telefonia móvel (Figura 3.2B);

Terceiro estágio: comunicação via satélite, utilizando modem IRIDIUM 9601 SBD (Figura 3.2C).

3.2 FAST

O projeto FAST [Alves & Cruz 2008] foi lançado pelo Departamento de Elétrica e de Computação da Faculdade de Engenharia da Universidade do Porto (FEUP), Portugal, em 2007, tendo como primeiro objetivo participar das competições Microtransat.

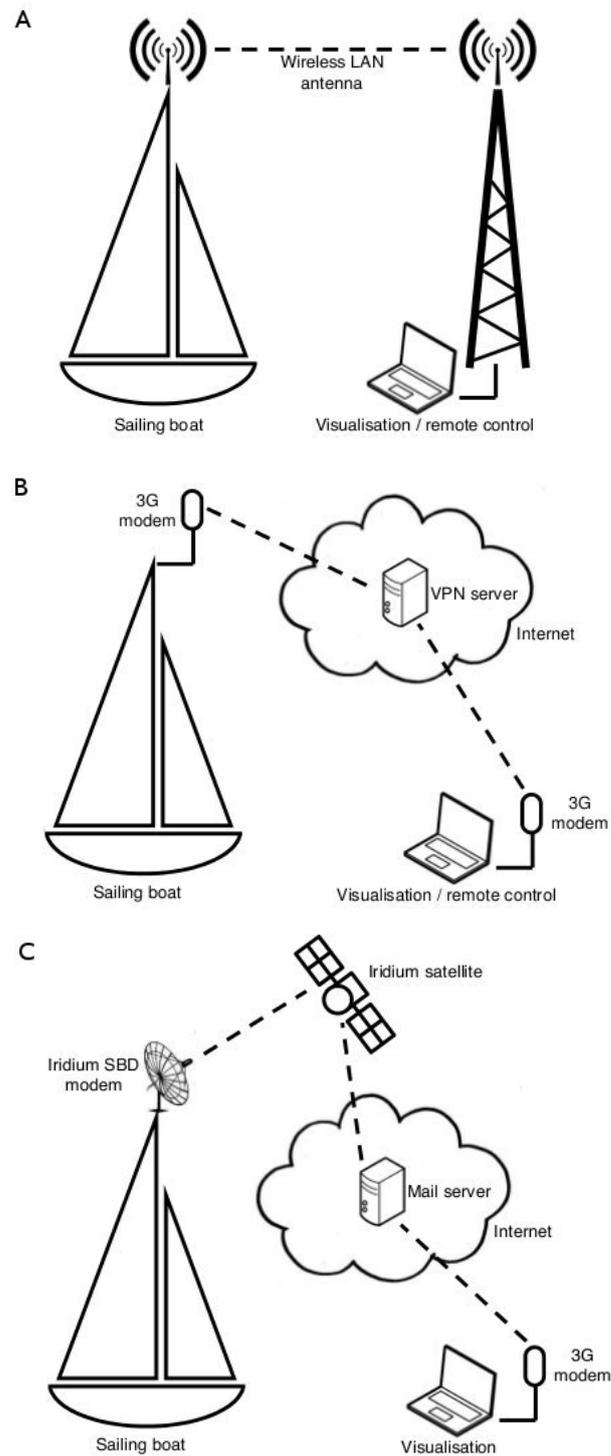


Figura 3.2: Sistema de comunicação de três estágios do ASV Robot. A) Primeiro estágio; B) Segundo estágio; C) Terceiro estágio.

Fonte: Stelzer & Jafarmadar (2009).

Seu sistema eletrônico é organizado em cinco subsistemas principais: computação, comunicação, sensores, atuadores e gerenciamento de energia. O sistema de computação é implementado em um pequeno computador de bordo baseado em um FPGA (*Field-Programmable Gate Array*) incluindo um microprocessador RISC, que fornece uma plataforma flexível de computação. O sistema de comunicação utiliza um roteador WiFi, para comunicação de curto alcance com um computador pessoal, um modem GSM, para comunicações de pequenos volumes de dados no mar a poucos quilômetros da costa, um modem IRIDIUM SBD, para comunicações a longas distâncias no mar em escala mundial, e um receptor de rádio controle, para controle totalmente manual da embarcação. Todos estes componentes são integrados como módulos de OEM e podem ser ligados e desligados sob o controle do *software* em execução no sistema de computação. Os sensores utilizados são: anemômetro, biruta, posição da retranca, bússola digital, GPS, inclinômetro, monitores de voltagem, sensor de luz ambiente, termômetro e um conjunto de sensores de água. O veleiro inclui três atuadores: dois servos de alta potência que controlam de forma independente dois lemes e um motor DC que controla as velas buja e a mestra simultaneamente. O subsistema de gerenciamento de energia inclui um painel solar de 45 Wp, duas baterias de íon-lítio de 95 Wh, um carregador de bateria, um módulo de gerenciamento e uma fonte de alimentação ultra eficiente.

O sistema operacional utilizado pelo FAST, em seu computador de bordo, é o μ CLinux¹, que é uma versão do sistema operacional Linux voltada para microcontroladores sem MMU. Ele fornece um console interativo de linha de comando, por meio de uma porta RS232 padrão, um sistema de arquivos estruturado, multitarefa e serviços TCP/IP básicos, como FTP, HTTP e TELNET.

Os dados coletados pelos sensores são entregues ao *software* de controle por meio de interfaces autônomas, após serem tratados. O *software* de controle é constituído por um conjunto de processos que se comunicam por *sockets* UDP. A sua arquitetura tenta imitar a sequência de comandos existente nas operações de barcos reais, que seguem uma hierarquia. Os comandos transitam do nível superior para o inferior como pacotes UDP.

O *software* é composto por cinco aplicativos: interface de *hardware*, *helm*, *sail*, *skipper* e *logger*. A interface de *hardware* implementa toda a comunicação com os sensores, atuadores e parâmetros de configuração. O *helm* controla os lemes, segundo algoritmos de controle e comandos providos do *skipper*. Manter um curso e executar as manobras básicas de cambagem e jibe, são exemplos desses comandos. O *sail* controla a posição das velas, com base na velocidade e direção do vento e um conjunto de regras que estabelecem a melhor angulação para estas. O *skipper* é o processo que ocupa o nível superior.

¹www.uclinux.org

Pela interface de *hardware* ele tem acesso a todas as informações coletadas pelos sensores. O *skipper* não tem ação direta sobre os atuadores, ficando esta sob a responsabilidade do *helm*, para os lemes, e o *sail*, para as velas. Ele também é responsável por decidir qual rumo tomar para chegar a um determinado ponto alvo, quando bordar ou jibar e comutar para o modo de controle manual ou autônomo. O *logger* é responsável por gravar todos os dados relevantes colhidos dos outros processos em um arquivo *log*.

3.3 Avalon

O veleiro autônomo Avalon foi construído por uma equipe da ETH-Zurique com o objetivo de cumprir o desafio principal da competição Microtransat, que é a travessia do Oceano Atlântico de forma totalmente autônoma [Erckens et al. 2010].

O Avalon utiliza como computador de bordo um MPC21 da DigitalLogic com 500 MHz, 1 GB de RAM e um disco rígido flash compacto. O veleiro é equipado com um GPS, uma unidade de medida inercial, um sensor de vento de ultra-som, que fornece a velocidade e direção do vento, e um sensor AIS (*Automatic Identification System*), que recebe como dados a posição e velocidade de outros barcos em VHF, e é utilizado no sistema anti-colisão. Para comunicação via satélite o veleiro utiliza um modem IRIDIUM 9522-B. A energia elétrica provem de um banco de baterias de lítio-manganês com capacidade total de 2,4 KWh alimentado por quatro painéis solares que produzem 90 Wp cada um. Uma célula de combustível de metanol é utilizada como fonte de energia de emergência. O veleiro utiliza um motor de 200 W para o controle da vela e dois motores de 150 W para controlar os lemes [Alt & Wittinghofer 2011].

A estrutura do *software* do Avalon se baseia em uma *Dynamic Data eXchange* (DDX), que é executada no Debian, uma distribuição do sistema operacional Linux. Esse *software* gerencia uma memória compartilhada chamada *store* que propociona uma forma de comunicação entre os subprogramas, que o formam, e que são executados em paralelo (Figura 3.3). Os *drivers* dos sensores coletam os dados lidos e os armazenam na memória compartilhada, na qual podem ser utilizados pelos programas de controle e de planejamento de curso [Erckens et al. 2010].

A arquitetura do *software* pode ser visualizada com detalhes na Figura 3.4. Os sensores utilizados pelo Avalon são: IMU, AIS e o WIND. A Unidade de Medidas Inercial (IMU) utilizada é, na realidade, uma combinação de uma IMU, um GPS e um barômetro e fornece como dados a posição, velocidade e aceleração, além da orientação e velocidade angular em três dimensões. O AIS fornece informações referentes a outras embarcações que estão navegando próximo ao veleiro. O sensor de vento (WIND) fornece a veloci-

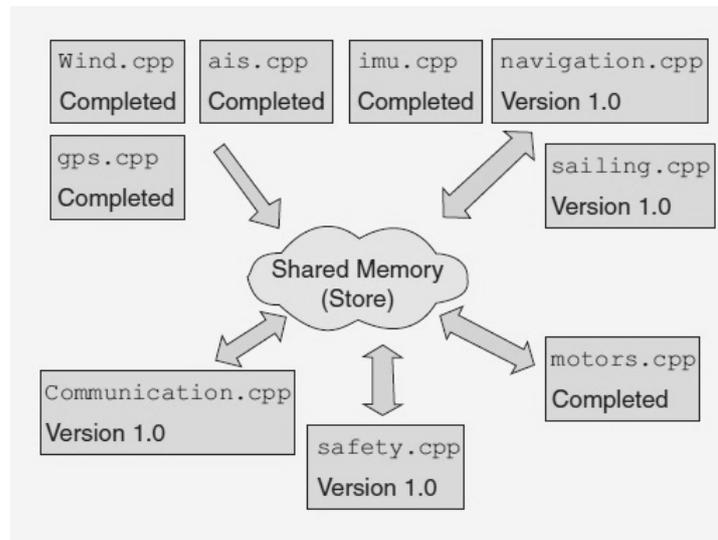


Figura 3.3: Organização do *software* do Avalon.
 Fonte: Erckens et al. (2010).

dade e direção do vento. O *destination_converter* tem como tarefa ler todos os pontos de destino contidos no arquivo texto *destination.txt* e escrevê-los na memória compartilhada. O *global_skipper* calcula o ponto de destino atual com base nas informações obtidas do IMU e nos pontos de destino armazenados. O *ais_eval* avisa ao *global_skipper* quando há a possibilidade de ocorrer uma colisão, fazendo com que ele calcule outro ponto de destino atual. O *navigator* calcula o melhor caminho entre a posição corrente e o ponto de destino atual para que o Avalon possa seguir. Ele gera diferentes *waypoints* e os armazena na memória compartilhada juntamente com os ângulos formados entre os *waypoints* consecutivos (Figura 3.5). O *skipper* gerencia os *waypoints* e decide qual posição desejada o Avalon precisa seguir, armazenando-a na memória compartilhada. O *sailor_statemachine* calcula os ângulos desejados dos lemes e da vela a partir da velocidade, posição atual e da posição desejada [Krucker 2010].

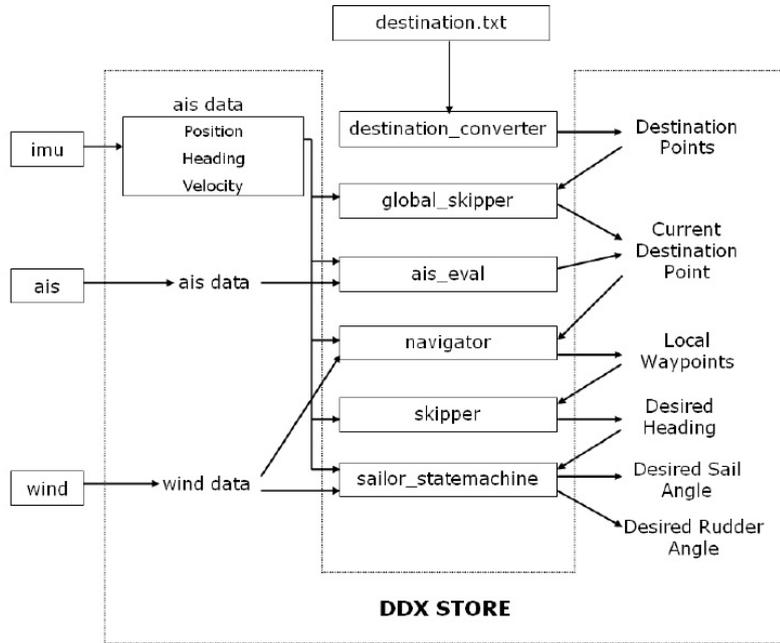


Figura 3.4: Diagrama da arquitetura do *software* do Avalon.
Fonte: Krucker (2010).

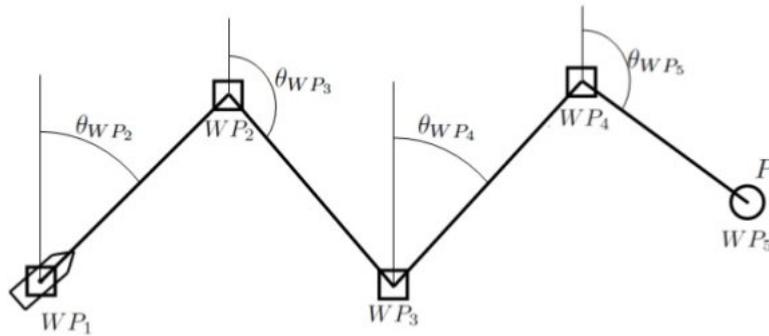


Figura 3.5: *Waypoints* calculados pelo *navigator*.
Fonte: Krucker (2010).

3.4 Dados técnicos dos veleiros autônomos estudados

As Figuras 3.6, 3.7 e 3.8 e as Tabelas 3.1, 3.2 e 3.3 apresentam, respectivamente, as imagens e alguns dados técnicos referentes aos veleiros autônomos ASV Robotat, Fast e Avalon.



Figura 3.6: ASV Robotat.

Fonte: Stelzer (2012).

Comprimento total	3,72 m
Boca	1,38 m
Pontal	0,55 m
Deslocamento	315 kg
Área vélica	4,5 m ²
Energia	3 painéis solares de 95 Wp Célula combustível de metanol
Baterias	Baterias de íon-lítio de 4,6 KWh
Computador	Mini ITX 800 MHz, 512 MB
Sistema operacional	Linux
Ano de construção	2006

Tabela 3.1: Dados do ASV Robotat.



Figura 3.7: FASt.
Fonte: Alves & Cruz (2008).

Comprimento total	2,50 m
Boca	0,67 m
Pontal	1,25 m
Deslocamento	50 kg
Área vélica	3,7 m ²
Energia	Painel solar de 45 Wp
Baterias	2 Baterias de íon-lítio de 95 Wh
Computador	Computador baseado em FPGA
Sistema operacional	μ CLinux
Ano de construção	2008

Tabela 3.2: Dados do FASt.



Figura 3.8: Avalon.
Fonte: Stelzer (2012).

Comprimento total	3,95 m
Boca	1,60 m
Pontal	0,25 m
Pontal (com quilha)	2,00 m
Deslocamento	450 kg
Área vélica	8,4 m ²
Energia	4 painéis solares de 90 Wp Célula combustível de metanol
Computador	MPC21 DigitalLogic 500 MHz, 1 GB
Baterias	Baterias de Li-Mn de 2,4 kWh
Sistema operacional	Debian Linux
Ano de construção	2009

Tabela 3.3: Dados do Avalon.

3.5 Comparação entre projetos

A tabela 3.4 mostra o resultado de um estudo comparativo dos principais componentes constituintes dos sistemas dos projetos pesquisados e do futuro veleiro autônomo N-Boat.

Projeto	ASV Roboat	FASt	Avalon	N-Boat
Propulsão a vento	x	x	x	x
Painel solar	x	x	x	x
Baterias	x	x	x	x
Receptor GPS	x	x	x	x
Anti-colisão			x	x
Combustível	x		x	
IMU			x	x
Propulsão elétrica				x
Sensor de nuvens				x
Estação meteorológica	x			x
<i>Payload</i> de parceiros				x
Bomba de drenagem	x			x
Câmera estéreo				x
<i>Laserscan</i>				x
Sensor de pressão	x		x	x
Comunicação: WiFi	x	x		x
Comunicação: VHF				x
Comunicação: Satélite	x	x	x	x
Comunicação: Rádio-modem				x
Comunicação: 3G	x			x
Comunicação: GPRS	x	x		x
Ultrassom (remoção de cracas)				x

Tabela 3.4: Comparação entre projetos de veleiros autônomos.

Até o momento em que este trabalho foi elaborado, os seguintes componentes já haviam sido obtidos para serem utilizados no protótipo de pesquisa:

- Arduino Uno e Mega;
- Raspberry Pi modelo B;
- baterias e carregador;
- receptor GPS;
- estação meteorológica; e
- módulo WiFi.

Capítulo 4

O Problema e a Solução

O problema aqui enfrentado é o desenvolvimento de uma arquitetura de *software* para a plataforma N-Boat, de modo a torná-la totalmente autônoma em suas missões. Tal arquitetura de *software* tem de gerenciar todos os módulos existentes no sistema, de forma a capacitar a plataforma a tomar decisões autonomamente. Do que foi dito, é esperado que a plataforma, quando em missão, determine o melhor rumo a ser tomado, tendo como base um ponto alvo, fornecido pela estação-base, e os dados coletados pelos seus sensores. Em sequência, escolher a melhor estratégia de navegação para seguir, bordejando ou jibando quando necessário, e sempre estar ciente da situação de sua segurança. Caso venha ocorrer algo que a comprometa, ter a habilidade de contornar o problema e retornar a um estado seguro de navegação.

No projeto da plataforma N-Boat é previsto equipá-lo com os componentes apresentados na Tabela 4.1.

Item	Componente
01	Propulsão a vento
02	Células fotovoltaicas / Painéis solares
03	Receptor GPS / Sensor de nuvens / Estação meteorológica
04	<i>Payload</i> de parceiros / Instrumentos científicos
05	Banco de baterias / Sistema de potência
06	Servos de dois graus de liberdade do controle de velas
07	Computador de bordo
08	Comunicação (WiFi, VHF, Satélite, Rádio-modem, 3G, GPRS)
09	Bomba de drenagem de água / Sirene
10	Propulsão elétrica alternativa
11	Servo do leme
12	Câmera estéreo
13	<i>Laserscan</i> / Ultrassom / Sensor de pressão
14	Outros instrumentos / IMU (unidade de medidas inerciais)

Tabela 4.1: Estimativa de componentes do veleiro N-Boat.

Tendo como ponto de partida o conhecimento dos módulos mínimos necessários para tornar uma plataforma totalmente autônoma, e os prováveis componentes que estarão a bordo do N-Boat, é sugerida a arquitetura de *software* apresentada na Figura 4.1 para a plataforma e na Figura 4.2 para a estação-base.

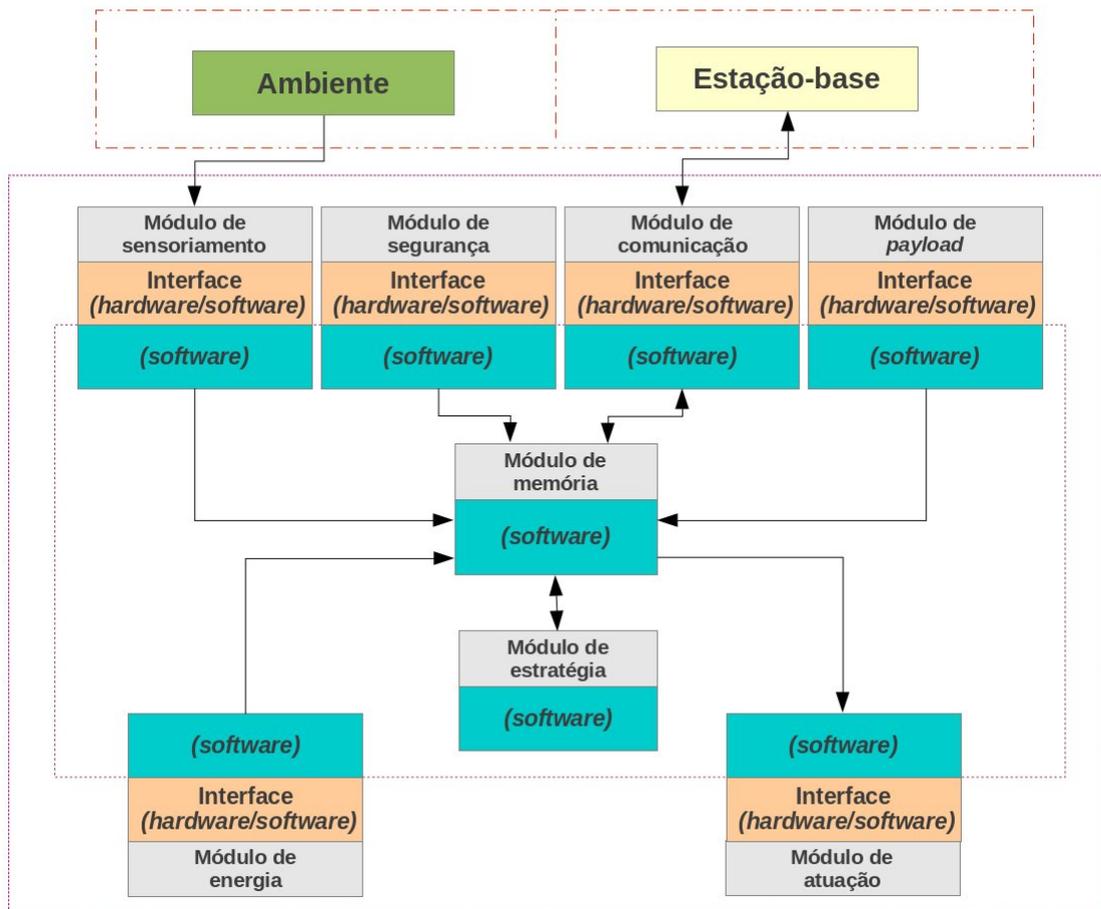


Figura 4.1: Diagrama da arquitetura de *software* da plataforma.

O *módulo de sensoriamento* capta do ambiente todas as informações necessárias à plataforma e as escreve no módulo de memória. O *módulo de comunicação* permite que a plataforma troque dados com a estação-base, realizando telemetria e recebendo missões. Esse módulo tanto pode escrever quanto ler no módulo de memória. O *módulo de energia* monitora os painéis solares e as baterias, podendo escrever no módulo de memória. O *módulo de segurança* é constituído por um sistema de anti-colisão e um sistema de detecção de mal tempo objetivando proporcionar ao N-Boat uma navegação segura. Esse módulo pode escrever no módulo de memória. Ambos os sistemas possuem recursos baseados em visão computacional. O *módulo de estratégia* é o mais importante. Ele é

responsável pela determinação do melhor rumo que a plataforma deve tomar, bem como a escolha da melhor manobra a ser seguida. É capaz de planejar a execução das manobras de bordejo e de jibe, e lê e escreve no módulo de memória. O *módulo de atuação* realiza o controle das velas e do leme. Também é responsável pelo controle dos componentes relacionados à propulsão elétrica alternativa. O *módulo de memória* é responsável pelo tratamento, armazenamento e gerenciamento de todos os dados providos dos outros módulos, permitindo a eles as operações de leitura e/ou escrita. O *módulo de payload* oferece a possibilidade de conectar ao sistema instrumentos de pesquisa diversos. Esse módulo escreve no módulo de memória para posterior transmissão dos dados via o módulo de comunicação.

Os *softwares* de todos os módulos apresentados acima, que compõem a arquitetura do sistema, são executados de forma concorrente no computador de bordo, estando localizados fora deste as suas respectivas interfaces de *hardware/software*, excetuando os módulos de memória e de estratégia, que não possuem interfaces.

Na Tabela 4.2 é apresentada a disposição dos componentes do N-Boat, listados na Tabela 4.1, nos módulos do sistema da plataforma.

Módulos	Componentes
Módulo de sensoriamento	Receptor GPS, sensor de nuvens, estação meteorológica, <i>laserscan</i> , ultrassom, sensor de pressão, IMU, outros instrumentos.
Módulo de segurança	Bomba de drenagem de água, sirene, câmera estéreo.
Módulo de <i>payload</i>	<i>Payload</i> de parceiros, instrumentos científicos.
Módulo de atuação	Propulsão a vento, servos de dois graus de liberdade do controle de velas, propulsão elétrica alternativa, servo do leme.
Módulo de comunicação	Comunicação (WiFi, VHF, satélite, rádio-modem, 3G, GPRS).
Módulo de energia	Células fotovoltaicas, painéis solares, banco de baterias, sistema de potência.
Módulo de memória	Computador de bordo.
Módulo de estratégia	Computador de bordo.

Tabela 4.2: Disposição dos componentes do N-Boat na arquitetura da plataforma.

Como ocorre na arquitetura da plataforma, o *módulo de comunicação* da arquitetura da estação-base permite que esta troque dados com a plataforma, recebendo telemetria e enviando missões. O módulo de *supervisão* é subdividido nos módulos de *visualização*,

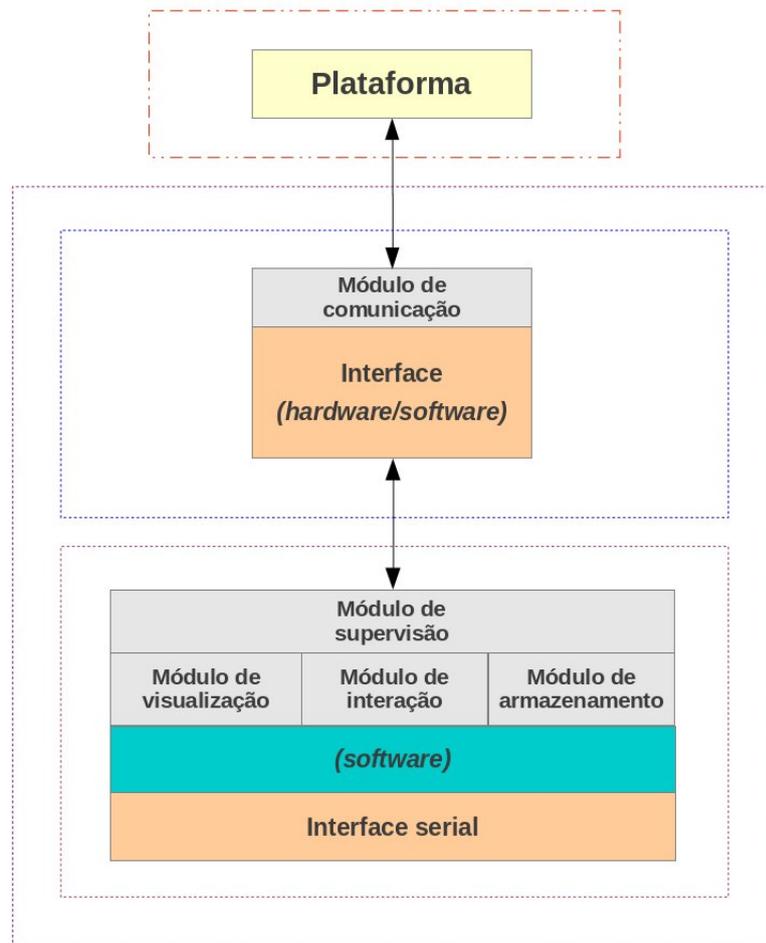


Figura 4.2: Diagrama da arquitetura de *software* da estação-base.

interação e *armazenamento* e dá todo o suporte que um sistema supervisório oferece, ou seja, apresentar os dados recebidos via telemetria, armazená-los e permitir interação com a plataforma, mediante envio de dados pelo módulo de comunicação.

O *software* do módulo supervisório, que engloba dos módulos de visualização, interação e armazenamento, é executado em um computador. O módulo de comunicação possui uma interface de *hardware/software* que possibilita a troca de dados entre a estação-base e a plataforma.

Capítulo 5

Implementação

Com o objetivo de validar a arquitetura proposta neste trabalho, foi desenvolvido um simulador na linguagem C, que utiliza a API gráfica OpenGL, sendo executado no sistema operacional Linux.

Na implementação do simulador, foram levados em conta os módulos da arquitetura apresentados nas Figuras 4.1 e 4.2, exetando os módulos de segurança e *payload*, presentes na plataforma. Tal procedimento foi adotado visando um futuro embarque do sistema no protótipo utilizado nas pesquisas, que é o nautimodelo RC exibido na Figura 1.1.

5.1 Simulador

O simulador desenvolvido é constituído por processos concorrentes que simulam os módulos de memória, estratégia, sensoriamento, energia, atuação e comunicação, presentes na plataforma, e os módulos de comunicação e supervisão, pertencentes à estação-base.

O processo que simula o módulo de memória cria uma memória compartilhada, cujo objetivo é possibilitar a comunicação entre os demais processos relacionados aos módulos pertencentes ao sistema da plataforma. Um semáforo binário controla o acesso a essa memória, permitindo que somente um processo por vez opere sobre ela, evitando o risco de utilização de dados parcialmente atualizados. Para que todos esses processos fossem gerenciados, foi implementado o algoritmo de controle apresentado na próxima subseção. O processo relacionado ao módulo de estratégia cria a rota que o veleiro deve navegar. Para isso, foi implementado o algoritmo de navegação descrito na subseção 5.1.2. O processo relacionado ao módulo de sensoriamento gera e escreve na memória compartilhada os seguintes dados: posição atual; velocidade e direção do barco; e velocidade e direção do vento. O processo que simula o módulo de energia gera dados acerca do nível de carga das baterias. O processo ligado ao módulo de atuação simula os ajustes dos ângulos

das velas e do leme, ao longo da navegação, para corrigir possíveis desvios e executar as manobras de bordejo ou de jibe, seguindo a rota traçada pelo algoritmo de navegação. O processo referente ao módulo de comunicação da plataforma lê, da memória compartilhada, dados relacionados à telemetria, fornecendo-os ao processo relacionado ao módulo de comunicação da estação-base, e deste recebendo os dados referentes à missão a ser cumprida, escrevendo-os na memória compartilhada. Finalmente, o processo relacionado ao módulo de supervisão lê os dados disponibilizados pelo processo ligado ao módulo de comunicação da estação-base e os exibe na tela.

5.2 Algoritmo de controle

O pseudo-código do algoritmo de controle é apresentado a seguir:

```
Algoritmo: Algoritmo de controle
01. Executa(software_módulo_de_memória)
02. Executa(software_módulo_de_sensoreamento)
03. Executa(software_módulo_de_comunicação)
04. Executa(software_módulo_de_estratégia)
05. Executa(software_módulo_de_atuação)
06. Executa(software_módulo_de_energia)
07. while verdadeiro do
08.   Escreve_Lê(software_módulo_de_sensoreamento,
09.             software_módulo_de_comunicação,
10.             software_módulo_de_estratégia,
11.             software_módulo_de_atuação,
12.             software_módulo_de_energia)
13. end while
```

Basicamente o algoritmo segue os seguintes passos:

1. O processo referente ao módulo de memória é executado (linha 1).
2. Os processos referentes aos demais módulos são executados, sendo a ordem apresentada não obrigatória (linhas 2 a 6).
3. Cada processo lê e/ou escreve de forma concorrente na memória compartilhada, sendo o seu acesso controlado por um semáforo binário (linhas 8 a 12).

5.3 Algoritmo de navegação

Para a implementação do algoritmo de navegação foi preciso realizar uma transformação de coordenadas sobre os dados geográficos de localização fornecidos pelo GPS

em dados que fossem possíveis de serem trabalhados em um plano de duas dimensões. As equações 5.1 e 5.2 [Stelzer & Pröll 2008] realizam a transformação requerida de forma aproximada, cujos resultados são aceitáveis para distâncias não tão grandes, pois não é levado em conta em sua modelagem a curvatura da Terra que se acentua para distâncias cada vez maiores. R_T é o raio da Terra, LAT e LON são latitude e longitude, respectivamente, em graus.

$$x = R_T \cdot \cos(LAT) \cdot \frac{\pi}{180^\circ} \cdot LON \quad (5.1)$$

$$y = R_T \cdot \frac{\pi}{180^\circ} \cdot LAT \quad (5.2)$$

O pseudo-código do algoritmo de navegação é apresentado a seguir:

```

Algoritmo: Algoritmo de navegação
01. while verdadeiro do
02.   LÊ(coordenadas_do_alvo)
03.   LÊ(coordenadas_do_barco)
04.    $d \leftarrow$  Calcula(distância)
05.    $da \leftarrow$  Calcula(direção_do_alvo)
06.    $dv \leftarrow$  LÊ(direção_do_vento)
07.    $dif \leftarrow \langle da - dv \rangle$ 
08.   if  $dif \geq 0$  and  $dif \leq 45$ 
09.      $f \leftarrow 45 - dif$ 
10.      $quant \leftarrow$  Calcula(pontos_e_subpontos_de_percurso[])
11.   else if  $dif > 45$  and  $dif \leq 150$ 
12.      $f \leftarrow 0$ 
13.      $quant \leftarrow$  Calcula(pontos_e_subpontos_de_percurso[])
14.   else
15.      $f \leftarrow dif - 150$ 
16.      $quant \leftarrow$  Calcula(pontos_e_subpontos_de_percurso[])
17.   end if
18.   contador  $\leftarrow 0$ 
19.   repeat
20.      $f \leftarrow -f$ 
21.     repeat
22.        $db \leftarrow$  LÊ(direção_do_barco)
23.        $dn \leftarrow da + f$ 
24.       if  $db \neq dn$ 
25.         Ajusta_velas( $dn$ )
26.         Ajusta_leme( $dn$ )
27.       end if
28.      $cb \leftarrow$  LÊ(coordenadas_do_barco)

```

```

29.   until  $cb = \text{pontos\_e\_subpontos\_de\_percurso}[\text{contador}]$ 
30.    $\text{contador} \leftarrow \text{contador} + 1$ 
31.    $dn \leftarrow da - f$ 
32.   Ajusta_velas( $dn$ )
33.   Ajusta_leme( $dn$ )
34.   until  $\text{contador} = \text{quant}$ 
35. end while

```

Os passos mais importantes do algoritmo são:

1. As coordenadas do alvo e as do barco são lidas nas linhas 2 e 3, respectivamente.
2. Nas linhas 4, 5 e 6 são obtidos os valores da distância entre o ponto inicial e o ponto alvo, direção em graus que o alvo faz com o norte e a direção em graus que o vento faz com o norte, respectivamente.
3. A diferença entre as direções do alvo-norte e vento-norte é calculada na linha 7.
4. Da linha 8 a 10 é tratada a situação de navegação contra o vento (bordejar). Da linha 11 a 13, a situação de navegação em rumo reto e da linha 14 a 16, a situação de navegação a favor do vento (jibar).
5. O laço que tem início na linha 19 e finaliza na linha 34 processa o controle da navegação do veleiro do ponto inicial até o ponto alvo.
6. O laço que se estende da linha 21 à linha 29 processa o controle da navegação em cada perna da navegação.
7. A condição que abrange as linhas 24 a 27 quando verdadeira propicia a realização do ajuste das velas e do leme quando ocorre desvios do rumo a ser seguido.
8. Nas linhas 31 a 33 é realizada a manobra de bordejo ou de jibe, caso a navegação seja contra ou a favor do vento, respectivamente.

5.4 Algoritmo de controle das velas

Para controlar as velas foi implementado o algoritmo apresentado mais adiante. O ponto chave do algoritmo é o polinômio que fornece o ângulo desejado para as velas a partir do ângulo da posição corrente do barco em relação ao vento. O polinômio foi calculado a partir de dados levantados experimentalmente utilizando o protótipo. Basicamente o experimento seguiu o seguinte procedimento:

- foram criadas marcas no chão orientadas pelos ângulos 45° , 80° , 115° , 150° e 180° ;
- posteriormente um ventilador foi fixado sobre a marca de ângulo 0° ;
- em seguida o nauti-modelo foi posicionado no centro das marcações com a proa voltada para a direção 0° ;

- o ventilador foi ligado e a posição do nauti-modelo foi modificada de forma a se alinhar com as marcas orientadas pelos ângulos 45° , 80° , 115° , 150° e 180° ;
- ao ser posicionado o nauti-modelo em cada ângulo, as escotas das velas foram puxadas e/ou folgadas de modo a fazerem as velas pararem de bater, obtendo assim a posição ideal das velas para aquela posição do vento;
- a cada posição do nauti-modelo foi anotado o seu ângulo em relação ao vento e o ângulo as velas, nas posições ideais, em relação ao nauti-modelo; e
- a partir dos ângulos levantados, o polinômio foi calculado utilizando o método de Lagrange.

O polinômio é apresentado abaixo e a Figura 5.1 mostra o seu gráfico gerado pelo Scilab.

$$P(x) = 37,682664275 - 3,351729835x + 0,081858239x^2 - 0,000655782x^3 + 0,000001741x^4$$

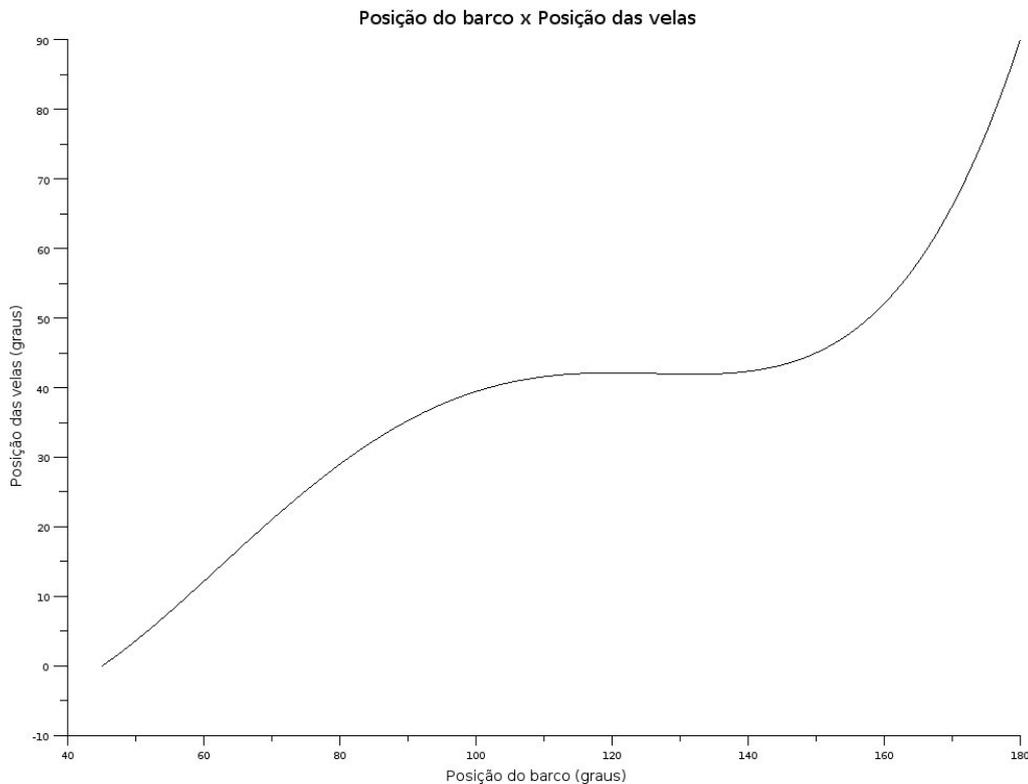


Figura 5.1: Gráfico do polinômio calculado, utilizado no controle das velas, que fornece a posição destas a partir da posição do barco em relação ao vento.

```

Algoritmo: Algoritmo de controle das velas
01. while verdadeiro do
02.    $pcv \leftarrow \text{Lê}(\text{posição\_corrente\_das\_velas})$ 
03.    $x \leftarrow \text{Lê}(dn)$ 
04.    $pdv \leftarrow 37,682664275 - 3,351729835*x + 0,081858239*x^2$ 
05.      $- 0,000655782*x^3 + 0,000001741*x^4$ 
06.   repeat
07.     if  $pcv > pdv$ 
08.        $pcv \leftarrow pcv - 0,1$ 
09.     end if
10.     if  $pcv < pdv$ 
11.        $pcv \leftarrow pcv + 0,1$ 
12.     end if
13.     Move_velas( $pcv$ )
14.   until  $\text{mod}(pcv - pdv) < LIMIAR$ 
15. end while

```

Os passos mais importantes do algoritmo são:

1. Na linha 2 é lida a posição corrente das velas em graus e na linha 3 é lida a direção de navegação do módulo de memória, sendo esta fornecida pelo módulo de estratégia.
2. A posição desejada das velas é calculada pelo polinômio na linha 4.
3. No laço da linha 6 à linha 14 é realizado o ajuste das velas. Caso o valor da posição corrente das velas for maior do que a posição desejada, o valor da posição corrente é decrementado de 0,1. Caso seja menor, o valor é incrementado de 0,1.
4. As velas são ajustadas efetivamente, de forma gradativa, mediante o valor corrigido da posição corrente das velas, na linha 13.

5.5 Algoritmo de controle do leme

Para controlar o leme foi implementado o algoritmo que segue abaixo.

```

Algoritmo: Algoritmo de controle do leme
01. while verdadeiro do
02.    $cont\_pspp \leftarrow \text{Lê}(\text{quantidade\_de\_pontos\_e\_subpontos\_de\_percurso})$ 
03.    $dnc \leftarrow \text{Lê}(dn)$ 
04.    $cont \leftarrow 0$ 
05.   repeat
06.     if  $dnc > 0$ 
07.       repeat
08.          $pl \leftarrow \text{Lê}(\text{posição\_do\_leme})$ 
09.         if  $pl < 60$ 

```

```

10.       $pl \leftarrow pl + 0,5$ 
11.      Move_leme( $pl$ )
12.      end if
13.       $pb \leftarrow L\hat{e}(posi\c{c}\tilde{a}o\_do\_barco)$ 
14.      until  $\text{mod}(pb - dnc) < LIMIAR$ 
15.      repeat
16.           $pl \leftarrow pl - 0,5$ 
17.          Move_leme( $pl$ )
18.      until  $pl < 0$ 
19.      end if
20.      if  $dnc < 0$ 
21.          repeat
22.               $pl \leftarrow L\hat{e}(posi\c{c}\tilde{a}o\_do\_leme)$ 
23.              if  $pl > -60$ 
24.                   $pl \leftarrow pl - 0,5$ 
25.                  Move_leme( $pl$ )
26.              end if
27.               $pb \leftarrow L\hat{e}(posi\c{c}\tilde{a}o\_do\_barco)$ 
28.              until  $\text{mod}(pb - dnc) < LIMIAR$ 
29.              repeat
30.                   $pl \leftarrow pl + 0,5$ 
31.                  Move_leme( $pl$ )
32.              until  $pl > 0$ 
33.              end if
34.               $m \leftarrow L\hat{e}(manobra)$ 
35.              if  $m = verdadeiro$ 
36.                   $dnc \leftarrow L\hat{e}(dn)$ 
37.                   $m \leftarrow falso$ 
38.                   $cont \leftarrow cont + 1$ 
39.              end if
40.          until  $cont = cont\_spp$ 
41.      end while

```

Os passos mais importantes do algoritmo são:

1. A quantidade de pontos e subpontos de percurso que serão atingidos é obtida na linha 1.
2. Na linha 2 a direção de navegação corrente é lida do módulo de memória, sendo esta fornecida pelo módulo de estratégia.
3. No laço principal, que se inicia na linha 5 e se estende até a linha 40, ocorre o controle do leme ao longo de toda a navegação. A partir do valor da direção de navegação corrente, a posição do leme é incrementado, linha 10, ou decrementado,

linha 24, de 0,5 graus. A restituição do leme para a posição 0 grau é realizada nas linhas 16 e 30.

4. O leme é ajustado efetivamente, de forma gradativa, mediante o valor corrigido da posição corrente do leme, nas linhas 11, 17, 25 e 31.
5. Na linha 34 é lido o valor da variável booleana *manobra* que armazena o valor lógico *verdadeiro* se ocorrer mudança de perna ou mantém o valor lógico *falso* caso contrário. Caso *manobra* seja *verdadeiro*, as instruções das linhas 36 a 39 serão processadas, sendo que na linha 36 o novo valor da direção de navegação será lido e na linha 37 o valor da variável *manobra* será *falso*.

Capítulo 6

Experimentos e Resultados

Visando a validação da arquitetura de *software* proposta, um simulador foi implementado, sendo este constituído de processos individuais que se comunicam mediante uma memória compartilhada controlada por um semáforo binário. A memória compartilhada foi implementada no módulo de memória juntamente com o algoritmo de controle. O algoritmo de navegação foi implementado no módulo de estratégia. Os algoritmos de controle das velas e do leme no módulo de atuação.

A fim de testar o simulador, vários experimentos foram realizados, simulando missões em diferentes distâncias, direções do vento e formas de navegação. Não foram consideradas nesta versão preliminar do simulador mudanças bruscas de direção e força do vento, forças de arrasto de correntezas ou qualquer tipo de atrito. Para apresentar os resultados obtidos da implementação da arquitetura proposta, uma simulação foi realizada utilizando as coordenadas geográficas de cinco pontos, sendo estas dispostas da seguinte forma:

- Ponto central: lat. $5^{\circ}50'34,675''$ S / lon. $35^{\circ}11'50,149''$ W (P_0);
- 1° quadrante: lat. $5^{\circ}49'53,840''$ S / lon. $35^{\circ}10'55,661''$ W (P_1);
- 2° quadrante: lat. $5^{\circ}49'36,530''$ S / lon. $35^{\circ}12'44,599''$ W (P_2);
- 3° quadrante: lat. $5^{\circ}51'58,541''$ S / lon. $35^{\circ}13'3,544''$ W (P_3);
- 4° quadrante: lat. $5^{\circ}51'44,767''$ S / lon. $35^{\circ}10'54,849''$ W (P_4).

Em cada quadrante foram praticadas as três formas de navegação, a saber, navegação em linha reta, bordejo (contra o vento) e jibe (a favor do vento). As Figuras 6.1, 6.2 e 6.3 e as Tabelas 6.2, 6.3 e 6.4 apresentam os resultados obtidos para o 1° quadrante, as Figuras 6.4, 6.5 e 6.6 e as Tabelas 6.5, 6.6 e 6.7 para o 2° quadrante, as Figuras 6.7, 6.8 e 6.9 e as Tabelas 6.8, 6.9 e 6.10 para o 3° quadrante e as Figuras 6.10, 6.11 e 6.12 e as Tabelas 6.11, 6.12 e 6.13 para o 4° quadrante. As Figuras de 6.1 a 6.12 apresentam imagens capturas durante a simulação. As Tabelas de 6.2 a 6.13 mostram os resultados dos seguintes parâmetros fornecidos pelo simulador ao longa da simulação:

- direção do vento-norte;
- direção do alvo-norte;
- direção de navegação;
- angulação das velas;
- quantidade de subpontos de percurso;
- distância em linha reta;
- distância entre pontos de percurso;
- distância de navegação entre pontos de percurso;
- distância de navegação total estimada; e
- afastamento máximo estimado da linha de rumo.

Nas figuras, o ponto de partida, o ponto de chegada, os pontos de percurso e os sub-pontos de percurso são representados, respectivamente, por quadrados nas cores branca, preta, verde e vermelha. A direção do vento é indicada por um triângulo vermelho. O veleiro é representado por um polígono amarelo.

Para verificar a precisão das distâncias calculadas pelo simulador do ponto P_0 aos pontos P_1 , P_2 , P_3 e P_4 , foi realizado um estudo comparativo, apresentado na Tabela 6.1, entre esses valores e os obtidos pela Fórmula de Haversine [Smart 1977] para os mesmos pontos. Tal fórmula é mais precisa devido esta levar em consideração em sua modelagem a curvatura da Terra.

Pontos	Distância (simulador)	Distância (haversine)	Erro
$P_0 - P_1$	2033,6499 m	2096,0 m	2,97%
$P_0 - P_2$	2532,2932 m	2455,0 m	3,15%
$P_0 - P_3$	3330,2092 m	3435,0 m	3,05%
$P_0 - P_4$	2837,7131 m	2752,0 m	3,11%

Tabela 6.1: Estudo comparativo das distâncias entre o ponto P_0 e os pontos $P_{1,2,3,4}$.

O simulador apresentou ao longo dos experimentos estabilidade no gerenciamento da memória compartilhada e no acesso a esta pelos processos referentes a cada módulo do sistema. As implementações dos algoritmos de controle, de navegação e de controle das velas e do leme também apresentaram bom desempenho, haja vista os resultados obtidos.

Em cada experimento, além dos pontos P_0 , P_1 , P_2 , P_3 e P_4 , foram aceitos como dados de entrada as direções do vento.

Nos experimentos 1a, 2a, 3a e 4a, os ângulos entre a direção do vento-norte e a direção do alvo-norte são $122,33^\circ$, $94,83^\circ$, $143,93^\circ$ e $102,72^\circ$, respectivamente. Esses ângulos caracterizam navegação em linha reta. Além de indicar esse tipo de navegação, o simulador forneceu todas as distâncias envolvidas no percurso a ser navegado, as coordenadas dos

pontos e subpontos de percurso, as direções de navegação a serem seguidas e a angulação que as velas devem assumir ao longo da navegação, como pode ser visto nas Tabelas 6.2, 6.5 e 6.8.

Nos experimentos 1b, 2b, 3b e 4b, os ângulos entre a direção do vento-norte e a direção do alvo-norte são, respectivamente, $17,33^\circ$, $14,83^\circ$, $8,93^\circ$ e $20,28^\circ$, que denotam navegação contra o vento. Acertadamente o simulador identificou o tipo de navegação a ser adotado e calculou todos os parâmetros de saída, apresentados nas Tabelas 6.3, 6.6 e 6.9.

Por fim, nos experimentos 1c, 2c, 3c e 4c, os ângulos entre a direção do vento-norte e a direção do alvo-norte são, respectivamente, $161,33^\circ$, $165,17^\circ$, $171,07^\circ$ e $184,72^\circ$, ângulos esses que indicam navegação a favor do vento. Nas Tabelas 6.4, 6.7 e 6.10 são mostrados os valores dos parâmetros calculados pelo simulador.

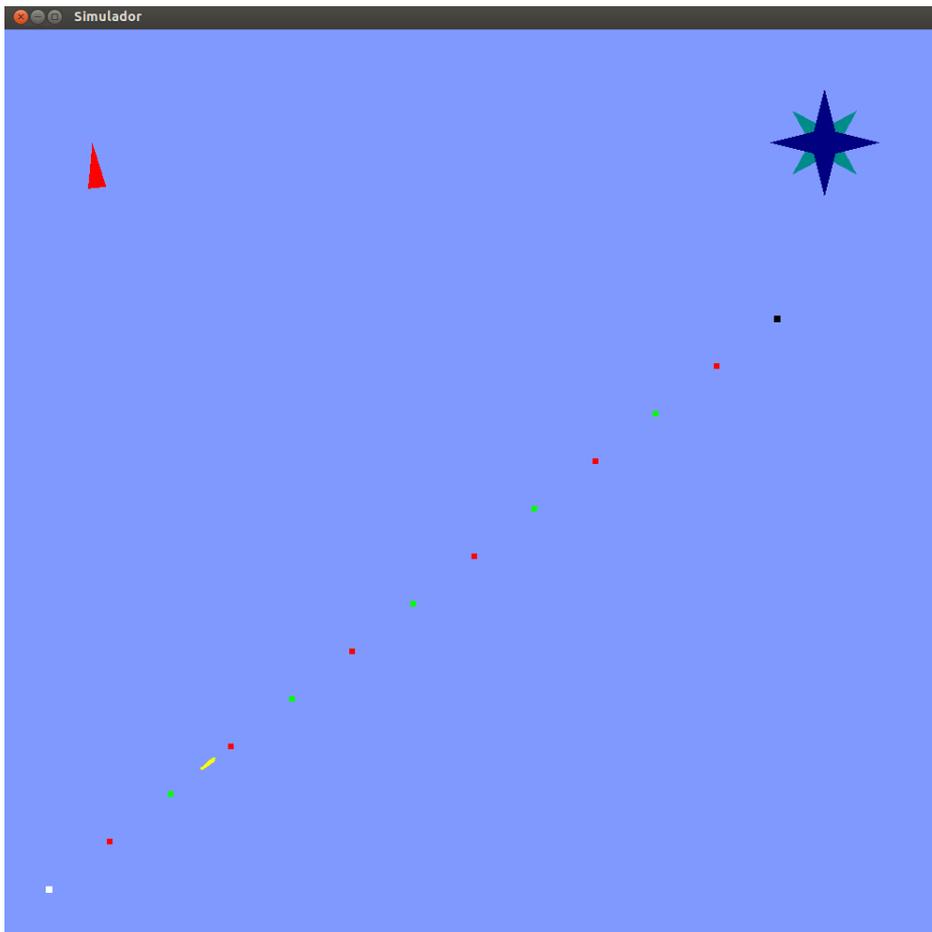


Figura 6.1: Experimento 1a: navegação em linha reta (1° quadrante).

Item calculado	Resultado
Direção do vento-norte	$174,0^{\circ}$
Direção do alvo-norte	$51,6677^{\circ}$
Direção de navegação	$51,6677^{\circ}$
Angulação das velas	$-42,0998^{\circ}$
Quantidade de subpontos de percurso	6
Distância em linha reta	2033,6499 m
Distância entre pontos de percurso	338,9417 m
Distância de navegação entre pontos de percurso	338,9417 m
Distância de navegação total estimada	2033,6499 m
Afastamento máximo estimado da linha de rumo	0,0 m

Tabela 6.2: Resultados referentes ao experimento 1a.

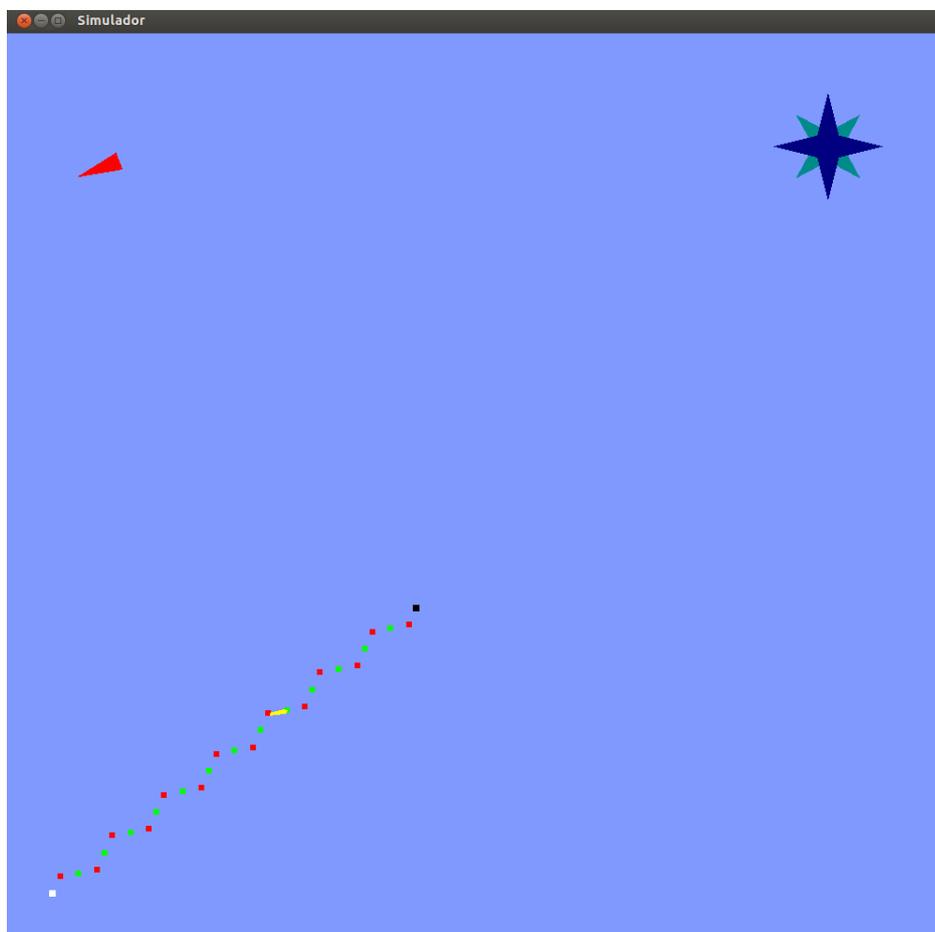


Figura 6.2: Experimento 1b: navegação contra o vento (1º quadrante).

Item calculado	Resultado
Direção do vento-norte	69,0°
Direção do alvo-norte	51,6677°
Direção de navegação	23,9999°/79,3354°
Angulação das velas	0,0°
Quantidade de subpontos de percurso	14
Distância em linha reta	2033,6499 m
Distância entre pontos de percurso	145,2607 m
Distância de navegação entre pontos de percurso	164,0149 m
Distância de navegação total estimada	2296,2092 m
Afastamento máximo estimado da linha de rumo	38,0796 m

Tabela 6.3: Resultados referentes ao experimento 1b.

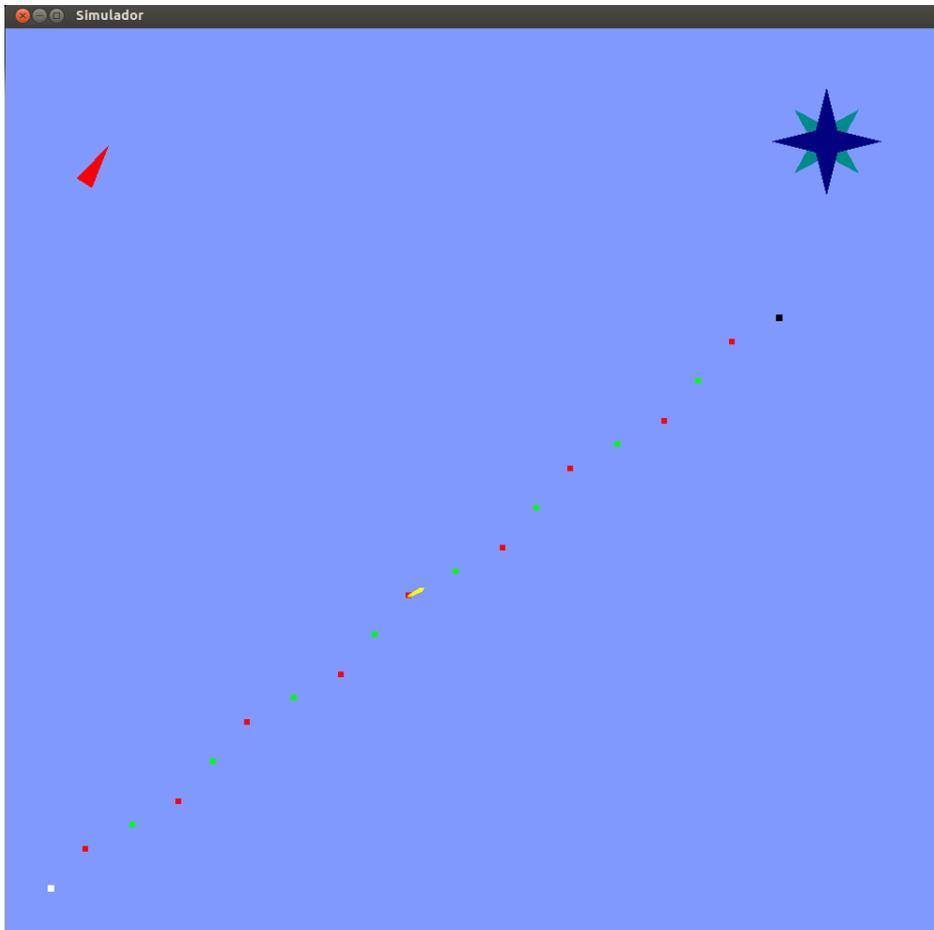


Figura 6.3: Experimento 1c: navegação a favor do vento (1° quadrante).

Item calculado	Resultado
Direção do vento-norte	-147,0°
Direção do alvo-norte	51,6677°
Direção de navegação	40,3354°/62,9999°
Angulação das velas	±53,4747°
Quantidade de subpontos de percurso	9
Distância em linha reta	2033,6499 m
Distância entre pontos de percurso	225,9611 m
Distância de navegação entre pontos de percurso	230,4540 m
Distância de navegação total estimada	2074,0859 m
Afastamento máximo estimado da linha de rumo	22,6419 m

Tabela 6.4: Resultados referentes ao experimento 1c.

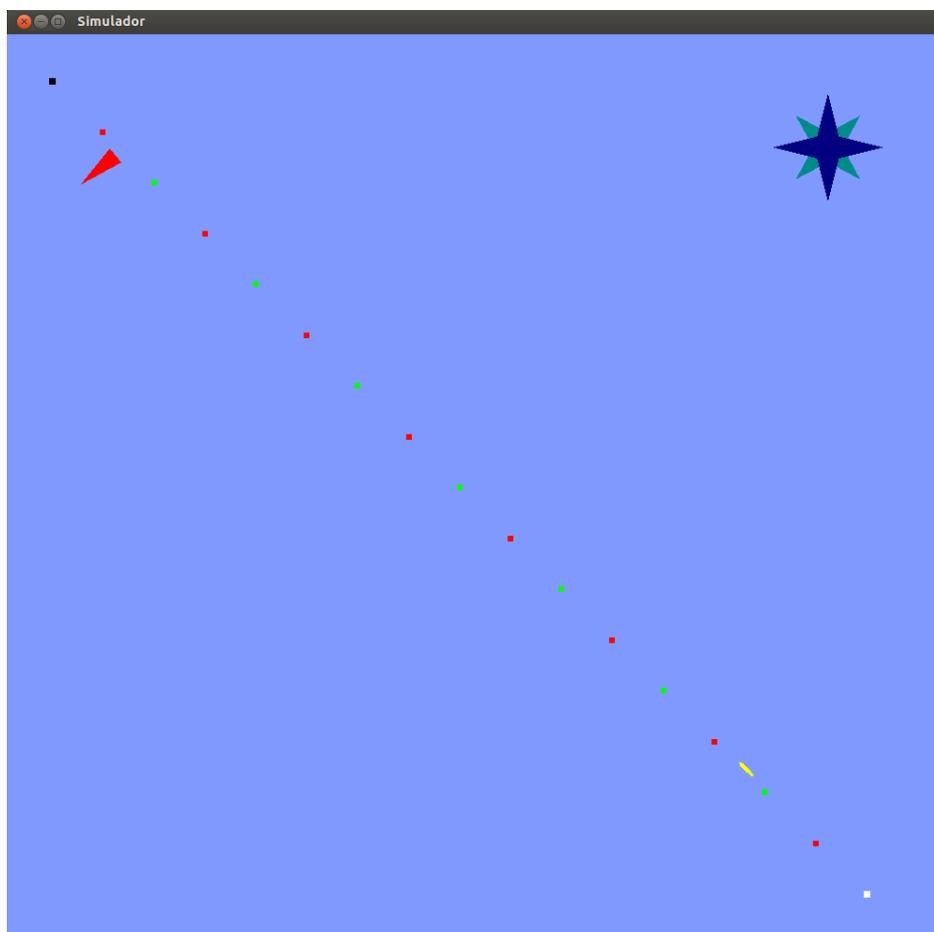


Figura 6.4: Experimento 2a: navegação em linha reta (2° quadrante).

Item calculado	Resultado
Direção do vento-norte	50,0°
Direção do alvo-norte	-44,8290°
Direção de navegação	-44,8290°
Angulação das velas	-37,5456°
Quantidade de subpontos de percurso	8
Distância em linha reta	2532,2932 m
Distância entre pontos de percurso	316,5367 m
Distância de navegação entre pontos de percurso	316,5367 m
Distância de navegação total estimada	2532,2932 m
Afastamento máximo estimado da linha de rumo	0,0 m

Tabela 6.5: Resultados referentes ao experimento 2a.

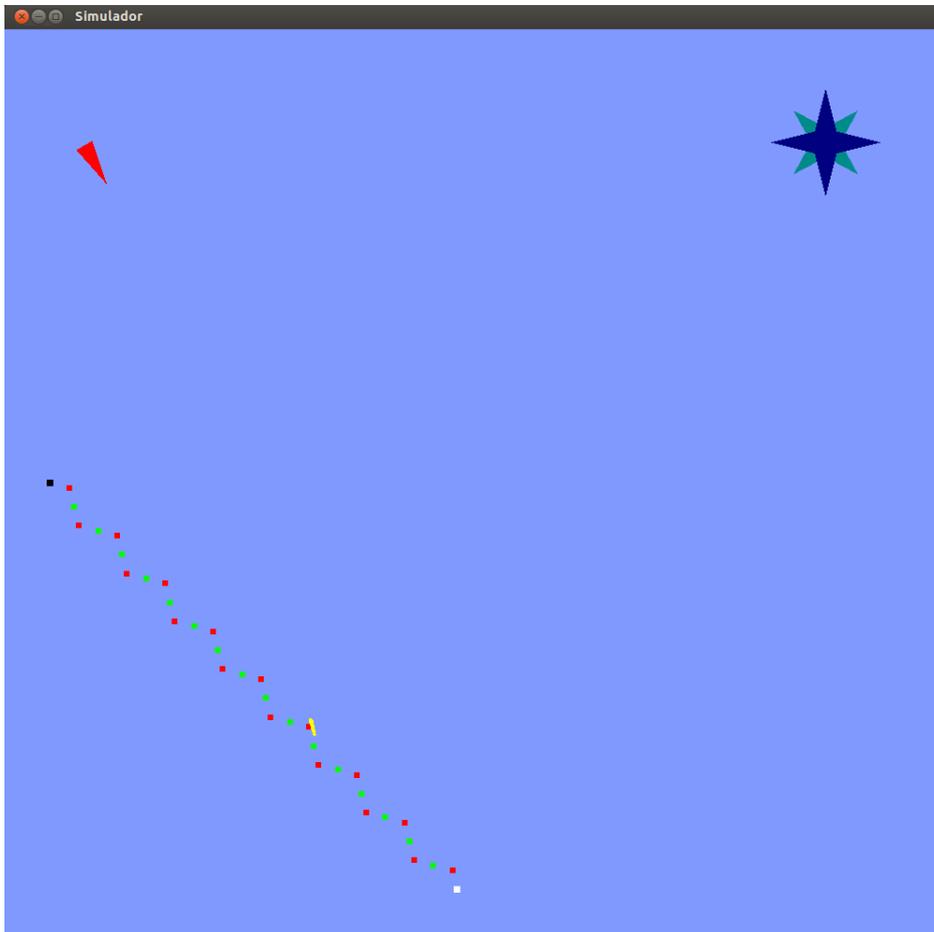


Figura 6.5: Experimento 2b: navegação contra o vento (2° quadrante).

Item calculado	Resultado
Direção do vento-norte	$-30,0^{\circ}$
Direção do alvo-norte	$-44,8290^{\circ}$
Direção de navegação	$-14,6580^{\circ}/-75,0000^{\circ}$
Angulação das velas	$0,0^{\circ}$
Quantidade de subpontos de percurso	17
Distância em linha reta	2532,2932 m
Distância entre pontos de percurso	148,9584 m
Distância de navegação entre pontos de percurso	172,3000 m
Distância de navegação total estimada	2929,1001 m
Afastamento máximo estimado da linha de rumo	43,2975 m

Tabela 6.6: Resultados referentes ao experimento 2b.

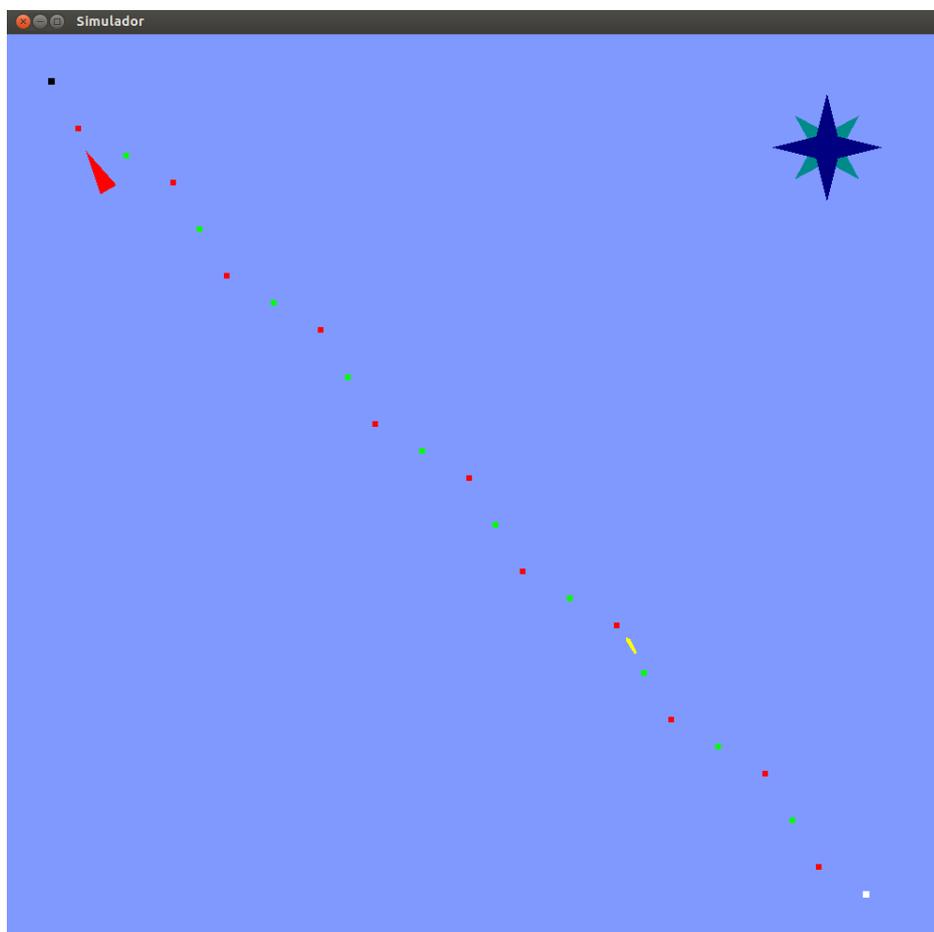


Figura 6.6: Experimento 2c: navegação a favor do vento (2º quadrante).

Item calculado	Resultado
Direção do vento-norte	150,0°
Direção do alvo-norte	-44,8290°
Direção de navegação	-60,0000°/-15,1709°
Angulação das velas	±58,2710°
Quantidade de subpontos de percurso	11
Distância em linha reta	2532,2932 m
Distância entre pontos de percurso	230,2085 m
Distância de navegação entre pontos de percurso	238,5211 m
Distância de navegação total estimada	2623,7327 m
Afastamento máximo estimado da linha de rumo	31,2106 m

Tabela 6.7: Resultados referentes ao experimento 2c.

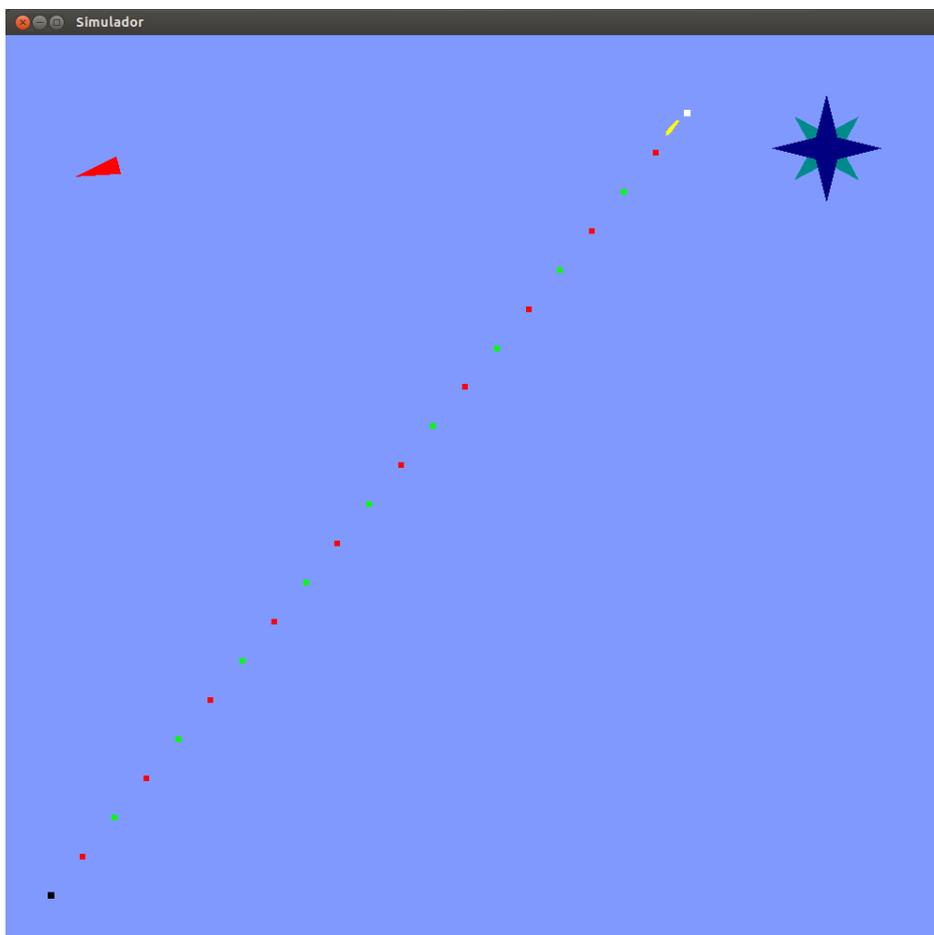


Figura 6.7: Experimento 3a: navegação em linha reta (3° quadrante).

Item calculado	Resultado
Direção do vento-norte	75,0°
Direção do alvo-norte	-141,0668°
Direção de navegação	-141,0668°
Angulação das velas	43,0014°
Quantidade de subpontos de percurso	10
Distância em linha reta	3330,2092 m
Distância entre pontos de percurso	333,0209 m
Distância de navegação entre pontos de percurso	333,0209 m
Distância de navegação total estimada	3330,2092 m
Afastamento máximo estimado da linha de rumo	0,0 m

Tabela 6.8: Resultados referentes ao experimento 3a.

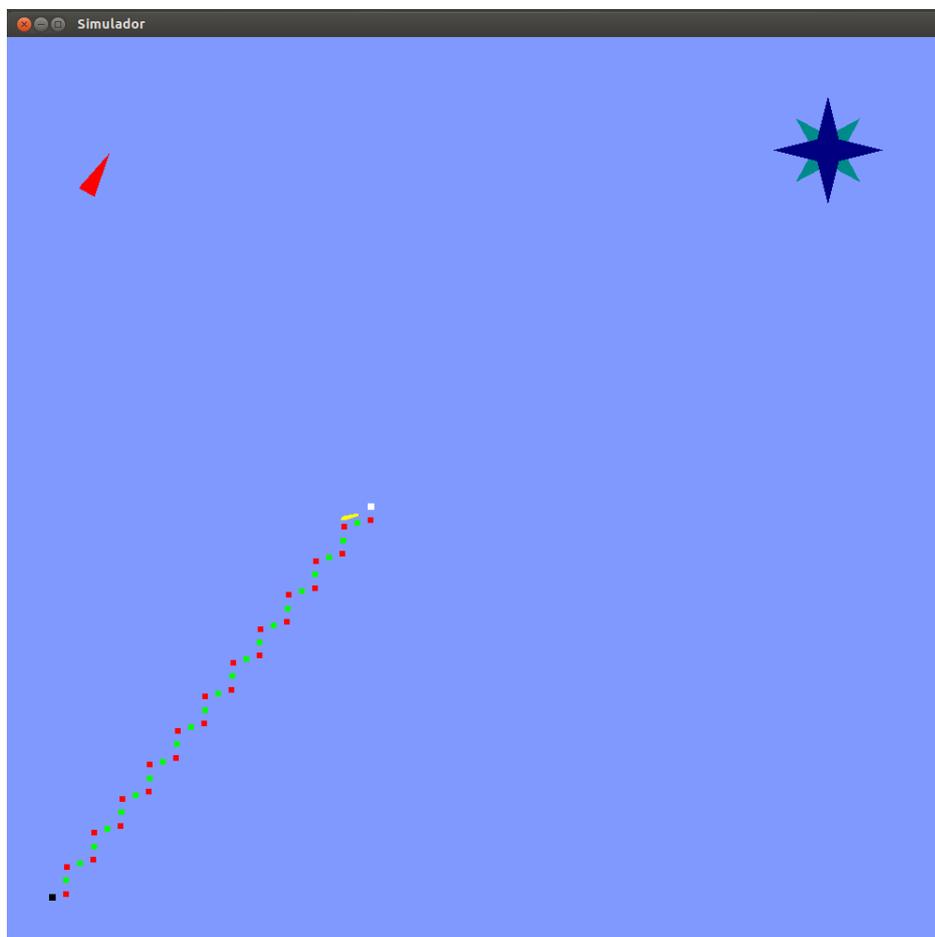


Figura 6.8: Experimento 3b: navegação contra o vento (3° quadrante).

Item calculado	Resultado
Direção do vento-norte	-150,0°
Direção do alvo-norte	-141,0668°
Direção de navegação	-165,0000°/-105,0000°
Angulação das velas	0,0°
Quantidade de subpontos de percurso	23
Distância em linha reta	3330,2092 m
Distância entre pontos de percurso	144,7917 m
Distância de navegação entre pontos de percurso	179,1242 m
Distância de navegação total estimada	4119,8569 m
Afastamento máximo estimado da linha de rumo	52,7277 m

Tabela 6.9: Resultados referentes ao experimento 3b.

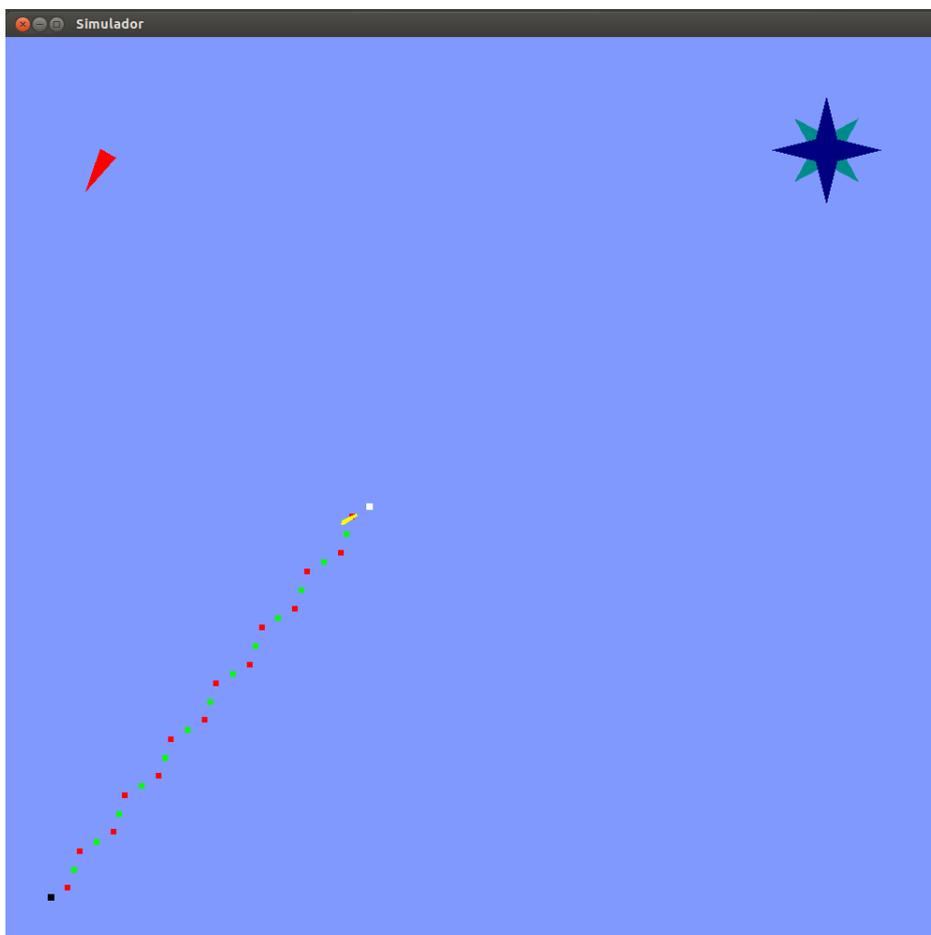


Figura 6.9: Experimento 3c: navegação a favor do vento (3° quadrante).

Item calculado	Resultado
Direção do vento-norte	30,0°
Direção do alvo-norte	-141,0668°
Direção de navegação	-120,0000°/-162,1336°
Angulação das velas	±68,1058°
Quantidade de subpontos de percurso	14
Distância em linha reta	3330,2092 m
Distância entre pontos de percurso	237,8721 m
Distância de navegação entre pontos de percurso	254,9097 m
Distância de navegação total estimada	3568,7363 m
Afastamento máximo estimado da linha de rumo	45,8144 m

Tabela 6.10: Resultados referentes ao experimento 3c.

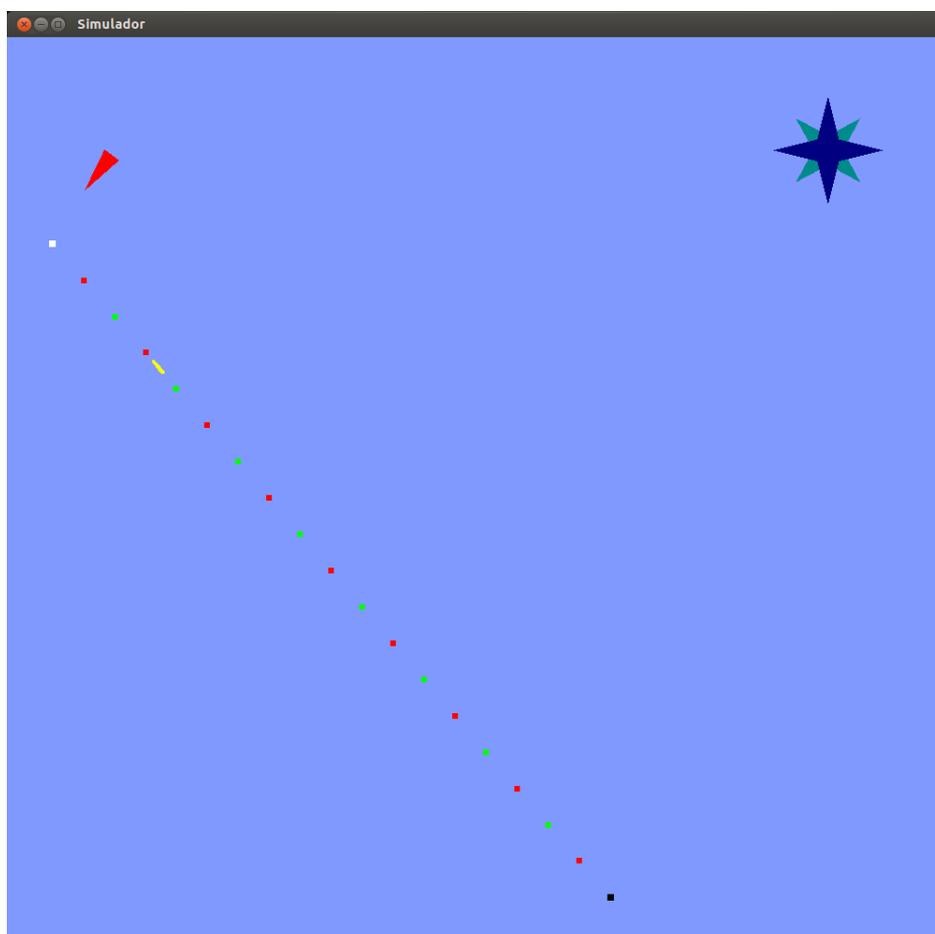


Figura 6.10: Experimento 4a: navegação em linha reta (4° quadrante).

Item calculado	Resultado
Direção do vento-norte	37,0°
Direção do alvo-norte	139,7239°
Direção de navegação	139,7239°
Angulação das velas	40,2121°
Quantidade de subpontos de percurso	9
Distância em linha reta	2837,7131 m
Distância entre pontos de percurso	315,3015 m
Distância de navegação entre pontos de percurso	315,3015 m
Distância de navegação total estimada	2837,7131 m
Afastamento máximo estimado da linha de rumo	0,0 m

Tabela 6.11: Resultados referentes ao experimento 4a.

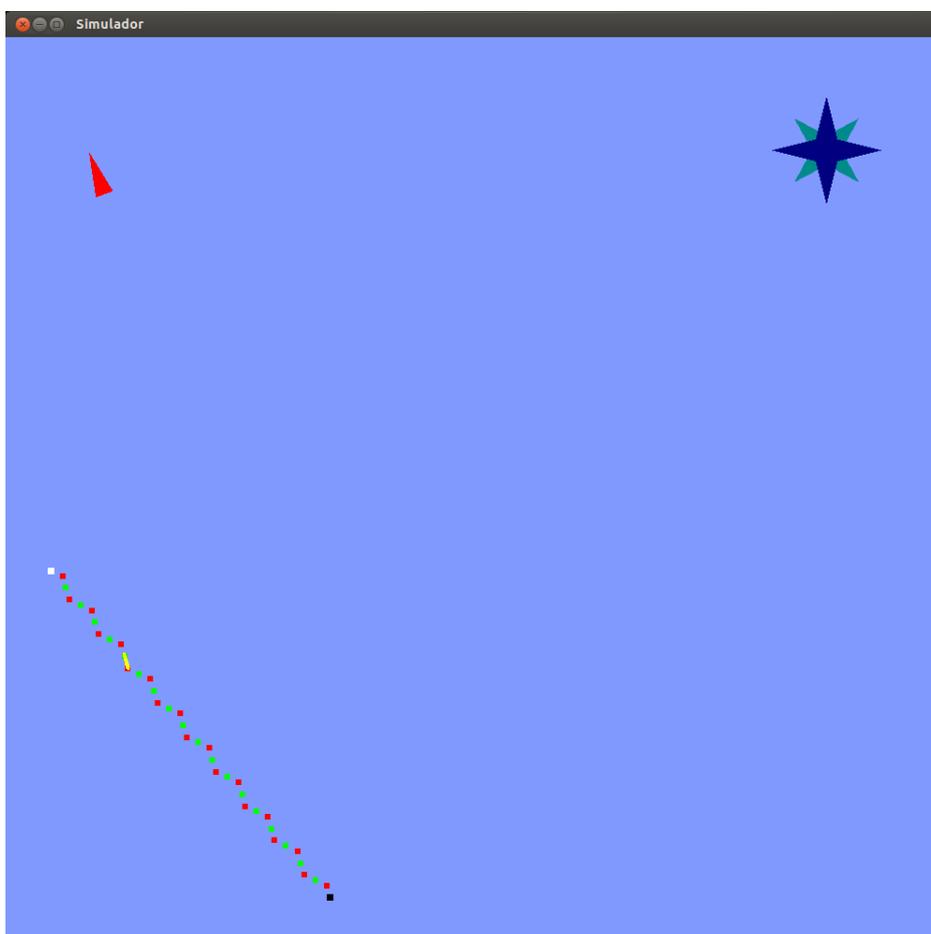


Figura 6.11: Experimento 4b: navegação contra o vento (4° quadrante).

Item calculado	Resultado
Direção do vento-norte	160,0°
Direção do alvo-norte	139,7239°
Direção de navegação	114,9999°/164,4478°
Angulação das velas	0,0°
Quantidade de subpontos de percurso	19
Distância em linha reta	2837,7131 m
Distância entre pontos de percurso	149,3533 m
Distância de navegação entre pontos de percurso	164,4256 m
Distância de navegação total estimada	3124,0867 m
Afastamento máximo estimado da linha de rumo	34,3852 m

Tabela 6.12: Resultados referentes ao experimento 4b.

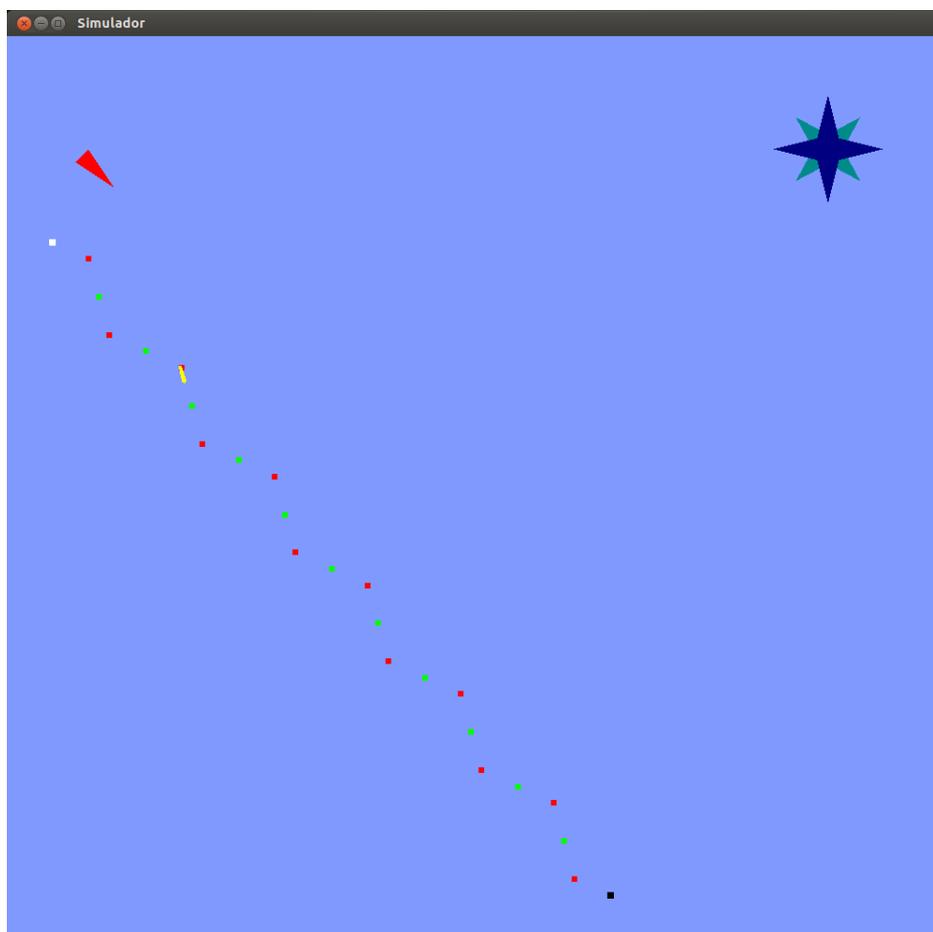


Figura 6.12: Experimento 4c: navegação a favor do vento (4° quadrante).

Item calculado	Resultado
Direção do vento-norte	-45,0°
Direção do alvo-norte	139,7239°
Direção de navegação	114,4478°/164,9999°
Angulação das velas	±77,2624°
Quantidade de subpontos de percurso	12
Distância em linha reta	2837,7131 m
Distância entre pontos de percurso	236,4761 m
Distância de navegação entre pontos de percurso	261,5131 m
Distância de navegação total estimada	3138,1575 m
Afastamento máximo estimado da linha de rumo	55,8305 m

Tabela 6.13: Resultados referentes ao experimento 4c.

Capítulo 7

Conclusões e Trabalhos Futuros

Neste trabalho foi proposta uma arquitetura de *software* para plataforma aquática autônoma. Tal proposta visa aplicar-se dentro do projeto N-Boat do laboratório Natalnet-DCA, que tem como objetivo principal capacitar um veleiro a navegar autonomamente.

A plataforma robótica, além de atuar de forma totalmente autônoma e segura, tem de interagir com a estação-base, enviando telemetria e recebendo desta as missões a serem cumpridas. A arquitetura de *software* sugerida visa, desta forma, tornar possível o gerenciamento desses requisitos. Ela é dividida em módulos, sendo cada um responsável por determinada tarefa dentro do sistema.

Para validar a arquitetura proposta, um simulador foi desenvolvido, sendo implementados os algoritmos de controle geral, navegação e controle das velas e do leme. Vários experimentos foram realizados e seus resultados mostraram-se satisfatórios, apesar de elementos importantes presentes em regime de operação real não terem sido levados em conta, tais como as variações da velocidade e da direção do vento e a força de arrasto das correntezas.

Como atividades futuras, pretende-se, primeiramente, implementar a arquitetura proposta no sistema de controle do protótipo de pesquisa, sendo testada e aprimorada para, em seguida, ser implementada no veleiro real a ser adquirido. Como uma primeira abordagem, os algoritmos de navegação e de controle das velas e do leme mostraram-se eficazes, mas apresentam limitações em suas funcionalidades. O algoritmo de navegação implementado não leva em conta as variações de direção e velocidade do vento ao longo da navegação nem tampouco é flexível o suficiente para ser utilizado em conjunto com o módulo de segurança. Para sanar essas limitações, é apresentada no Apêndice B uma versão melhorada deste algoritmo. O polinômio utilizado no algoritmo de controle das velas pode ser mais preciso se calculado a partir de dados experimentais levantados de forma mais rigorosa. Seus resultados também podem vir a serem adotados como pontos preliminares para um sistema de controle mais avançado como, por exemplo, um contro-

lador *Fuzzy*. O controle do leme pode ganhar em desempenho implementando-o como um controlador PID.

Referências Bibliográficas

- Alt, Christian & Natalie Wittinghofer (2011), Autonomous sailing boats, Seminar aus informatik, Universität Salzburg, Áustria.
- Alves, José C. & Nuno A. Cruz (2008), ‘Fast – an autonomous sailing platform for oceanographic missions’, *IEEE Conference Publications*, pp. 1-7 .
- Briere, Yves (2008), ‘Iboat: An autonomous robot for long-term offshore operation’, *IEEE Conference Publications*, pp. 323-329 .
- Dent, Nicholas (1978), *Velejar – Manual prático de técnicas e equipamentos*, Editora Abril, São Paulo, SP.
- Erckens, Hendrik, Gion-Andri Büsser, Cédric Pradalier & Roland Y. Siegwart (2010), ‘Avalon – navigation strategy and trajectory following controller for an autonomous sailing vessel’, *IEEE Journals & Magazines* 17(1), pp. 45-54 .
- Francisco, Portal São (2013), Página na internet. Brasil. Acesso em out/2013. Disponível em: <http://www.portalsaofrancisco.com.br/alfa/barco-a-vela/barco-a-vela-5.php>.
- Hobbycontrol (2013), Página na internet. Brasil. Acesso em out/2013. Disponível em: <http://www.hobbycontrol.com.br/details/CAISER-01/monsoon-900-branco>.
- INCT-SEC (2013), Página na internet., INCT-SEC – Instituto Nacional de Ciência e Tecnologia em Sistemas Embarcados Críticos, ICMC, USP. Brasil. Acesso em out/2013. Disponível em: <http://www.inct-sec.org/br/aplicacoes/veiculos-nauticos-autonomos>.
- Jaulin, Luc & Fabrice Le Bars (2013), ‘An interval approach for stability analysis: Application to sailboat robotics’, *IEEE Journals & Magazines* 29(1), pp. 282-287 .
- Krucker, Mario (2010), Simulation, control and testing of an autonomous sailboat, Dissertação de mestrado, ETH, Zurique, Suíça.

- Microtransat (2013), The microtransat challenge, Página na internet. Acesso em out/2013. Disponível em: <http://www.microtransat.org>.
- Miguens, Altineu Pires (2000), *Navegação: A Ciência e a Arte. Vol. I: Navegação Costeira, estimada e em Águas Restritas. Cap. 11: Instrumentos Náuticos*, Marinha do Brasil.
- Miller, Paul, Owen Brooks & Matthew Hamlet (2009), ‘Development of the usna sailbots (asv)’, *Proceedings of the 2nd International Robotic Sailing Conference* . Matosinhos, Portugal.
- QMAST (2013), Queen’s mostly autonomous sailboat, Página na internet, Ontário, Canadá. Acesso em out/2013. Disponível em: <http://qmast.ca/>.
- Sauzé, Colin (2005), Control software for a sailing robot, Dissertação de mestrado, University of Wales, Aberystwyth.
- Schmidt, João G. (1979), *Aprenda a Velejar*, Ediouro S.A., Rio de Janeiro, RJ.
- Sliwka, Jan, Pierre-Henri Reilhac, Richard Leloup, Pierre Crepier, Henry De Malet, Patrick Sittaramane, Fabrice Le Bars, Kostia Roncin, Bruno Aizier & Luc Jaulin (2009), ‘Autonomous robotic boat of ensieta’, *Proceedings of the 2nd International Robotic Sailing Conference* . Matosinhos, Portugal.
- Smart, W. M. (1977), *Textbook on Spherical Astronomy*, Cambridge University Press.
- Stelzer, Roland (2012), Autonomous Sailboat Navigation – Novel Algorithms and Experimental Demonstration, Tese de doutorado, De Montfort University, Leicester.
- Stelzer, Roland & Karim Jafarmadar (2007), ‘A layered system architecture to control an autonomous sailboat’, *TAROS 2007, Aberystwyth, UK* .
- Stelzer, Roland & Karim Jafarmadar (2009), ‘Communication architecture for autonomous sailboats’, *Proceedings of the 2nd International Robotic Sailing Conference* . Matosinhos, Portugal.
- Stelzer, Roland & Tobias Pröll (2008), ‘Autonomous sailboat navigation for short course racing’, *Robotics and Autonomous Systems* **56**(7), pp. 604-614 .

Apêndice A

Principais termos náuticos

Abaixo seguem alguns dos principais termos náuticos utilizados em veleiros obtidos, em sua maior parte, em [Schimidt 1979]:

Abatimento: Desvio lateral do rumo seguido ocasionado pela ação da correnteza e/ou do vento.

Adernamento: Inclinação para bombordo ou estibordo que uma embarcação pode estar sujeita.

Alhetas: São os cantos da popa.

Amuras: São as duas regiões mais encurvadas próximas à proa.

Área vélica: Área total das velas.

Barlavento: Direção de onde vem o vento.

Boca: É a maior largura do casco.

Bolina: Peça laminar móvel de madeira ou de metal utilizada em barcos pequenos para criar resistência lateral de modo a minimizar o abatimento.

Borda: Contorno superior do casco.

Bordo livre: Ou costado, é a parte do casco que se encontra entre a borda e a linha de flutuação.

Calado: Parte do casco que se encontra entre a linha de flutuação e a face inferior da quilha.

Comprimento de fora a fora: É o comprimento do barco.

Corrida da milha: É um procedimento utilizado com o objetivo de averiguar a exatidão das indicações dos odômetros e dos velocímetros, a fim de executar, caso seja necessária, a calibração destes. Ele é realizado fazendo-se o navio percorrer uma distância conhecida (geralmente 3 ou mais milhas) em um sentido e no sentido oposto, para anular a influência da corrente e do vento, e medindo o tempo gasto em cada percurso, tendo-se, assim, as respectivas velocidades.

Derrota: Rumo seguido no mar por uma embarcação.

Deslocamento: O volume de água que um barco desloca, cujo peso é idêntico ao peso do barco.

Linha de flutuação: Ou linha-d'água, é a linha que separa as obras vivas das obras mortas.

Mar fechado: Mar que não possui ligação com oceanos.

Mediana: É o eixo simétrico que divide o casco longitudinalmente em duas metades.

Meia-nau: É o eixo que divide o casco a meio comprimento transversalmente.

Milha marítima: Medida de comprimento que equivale à 1.852 m.

Navegação em águas restritas: Navegação praticada a até 3 milhas náuticas da costa.

Navegação costeira: Navegação praticada a distâncias que variam de 3 a 50 milhas náuticas da costa.

Navegação oceânica: Navegação realizada em alto-mar a mais de 50 milhas náuticas da costa.

Nó: Medida de velocidade que equivale à uma milha marítima por hora, ou 1,852 km/h.

Obras mortas: É a parte do casco que fica fora da água.

Obras vivas: Ou carena, é a parte imersa do casco.

Perna: Ou bordo, é cada zigue e cada zague realizado quando se bordeja ou jiba.

Pontal: É a altura total do casco (calado + bordo livre).

Por ante-a-ré: Localização mais próxima da popa.

Por ante-a-vante: Localização mais próxima da proa.

Proeiro: Tripulante de um veleiro que manobra a buja e a bolina.

Sotavento: Direção para onde sopra o vento.

Timoneiro: Tripulante de um veleiro que manobra a vela mestra e o leme.

Apêndice B

Algoritmo de navegação modificado

Algoritmo: Algoritmo de navegação modificado

01. **while true do**
02. Lê(*coodenadas_do_alvo*)
03. Lê(*coodenadas_do_barco*)
04. $d \leftarrow \text{Calcula}(\textit{distância})$
05. $\textit{quant_pp} \leftarrow \text{Calcula}(\textit{pontos_de_percurso}[i])$
06. $\textit{contador_pp} \leftarrow 0$
07. **repeat**
08. $\textit{dv} \leftarrow \text{Lê}(\textit{direção_do_vento})$
09. $\textit{dpp} \leftarrow \text{Calcula}(\textit{direção_do_ponto_de_percurso})$
10. $\textit{dif} \leftarrow \langle \textit{dv} - \textit{dpp} \rangle$
11. **if** $\textit{dif} \geq 0$ **and** $\textit{dif} \leq 45$
12. $f \leftarrow 45 - \textit{dif}$
13. $\textit{quant_spp} \leftarrow \text{Calcula}(\textit{subpontos_de_percurso}[i])$
14. $\textit{contador_spp} \leftarrow 0$
15. **else if** $\textit{dif} > 45$ **and** $\textit{dif} < 150$
16. $f \leftarrow 0$
17. $\textit{quant_spp} \leftarrow \text{Calcula}(\textit{subpontos_de_percurso}[i])$
18. $\textit{contador_spp} \leftarrow 0$
19. **else**
20. $f \leftarrow \textit{dif} - 150$
21. $\textit{quant_spp} \leftarrow \text{Calcula}(\textit{subpontos_de_percurso}[i])$
22. $\textit{contador_spp} \leftarrow 0$
23. **end if**
24. **repeat**
25. $f \leftarrow -f$
26. **repeat**
27. $\textit{db} \leftarrow \text{Lê}(\textit{direção_do_barco})$
28. $\textit{dn} \leftarrow \textit{dpp} + f$
29. **if** $\textit{db} \neq \textit{dn}$

```

30.     Ajusta_velas(dn)
31.     Ajusta_leme(dn)
32.     end if
33.      $cb \leftarrow L\hat{e}(coordenadas\_do\_barco)$ 
34.     until  $cb = subpontos\_de\_percurso[contador\_spp]$ 
35.      $contador\_spp \leftarrow contador\_spp + 1$ 
36.     until  $contador\_spp = quant\_spp$ 
37.      $contador\_pp \leftarrow contador\_pp + 1$ 
38.      $dn \leftarrow dpp - f$ 
39.     Ajusta_velas(dn)
40.     Ajusta_leme(dn)
41.     until  $contador\_pp = quant\_pp$ 
42. end while

```

Os passos mais importantes do algoritmo são:

1. As coordenadas do alvo e as do barco são lidas nas linhas 2 e 3, respectivamente.
2. Na linha 4 é calculada a distância do ponto de partida até o ponto de chegada.
3. Na linha 5 os pontos de percurso são calculados, bem como a sua quantidade.
4. No laço principal, que compreende o trecho de vai da linha 7 até a linha 41, ocorre o processamento que abrange toda a navegação.
5. Ao contrário do que acontece no algoritmo original, toda vez que ocorrer uma mudança de perna, é verificado se a direção do vento variou, sendo novamente avaliado qual o tipo de navegação que será seguido, se contra ou a favor do vento, ou em rumo reto.
6. Nas linhas 30 e 31 são realizados ajustes nas velas e no leme caso ocorra desvio no rumo de navegação traçado, sendo esse corrigido. Nas linhas 39 e 40 as velas e o leme são ajustados para ser realizada uma manobra de bordejo ou de jibe, sendo a direção de navegação utilizada no módulo de atuação.