

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Rover submarino

José Tiago Cardoso Meireles

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Bruno Miguel Mateus Ferreira

Co-orientador: Prof. Aníbal Castilho Coimbra de Matos

25 de Junho de 2017

Resumo

Esta dissertação descreve o trabalho realizado no desenvolvimento um rover submarino capaz de se movimentar no solo subaquático. O *Small Hovering AUV with Differential actuation* (SHAD) foi considerado a base sobre a qual se realizou a adaptação necessária para o desenvolvimento do rover.

A oferta no mercado para *Autonomous Underwater Vehicles* (AUV) capazes de operar perto e ou no solo é escassa. Existem ainda os *Remotely Operated Vehicles* (ROV), veículos que são tipicamente grandes, com formas pouco hidrodinâmicas e que operam a baixa velocidade. Com estes aspetos em mente, foi pensada uma solução capaz de se deslocar no solo com uma forma hidrodinâmica, capaz de ser implementada em AUV.

Inicialmente foi realizado um levantamento do estado da arte, que serviu de base para o restante trabalho desenvolvido. Foram escolhidos os componentes necessários e ainda foi escolhida a configuração da solução para o deslocamento sobre superfícies subaquáticas. Todos os componentes foram implementados num protótipo que posteriormente foi sujeito a um conjunto de testes para validação do sistema.

Com este trabalho foi possível desenvolver um veículo autónomo capaz de se deslocar sobre uma superfície subaquática com a capacidade de calcular a sua localização relativa.

Abstract

This dissertation describes the work carried out with the objective of developing an underwater rover able to move in the underwater surfaces. The Small Hovering AUV with Differential actuation (SHAD) was considered the basis on which the necessary adaptations were made for the development of the rover.

The market offer for Autonomous Underwater Vehicles (AUV) capable of operating close to or on the ground is scarce. There are also Remotely Operated Vehicles (ROV), vehicles that are typically large, not hydrodynamic, and operate at low speeds. With these aspects in mind, a solution capable of traveling in the ground with a hydrodynamic form, capable of being implemented in AUV was devised.

Initially a survey of the state of the art was carried out, which served as the basis for the rest of the work developed. The required components were chosen and the configuration of the solution to allow the robot to move on underwater surfaces was selected. All components were implemented in a prototype that was later subjected to a set of tests for systems validation.

With the work carried out it was possible to develop an autonomous vehicle capable of moving on an underwater surface with the ability to calculate its relative location.

Agradecimentos

Quero começar por agradecer à minha família, em particular à minha mãe e ao meu pai, pelos esforços realizados nos últimos 5 anos. Ainda uma palavra para a minha irmã, por estar disposta a trabalhar por mim em casa. Quero lembrar também os meus avós que já partiram e ainda a minha avó que apesar de já não estar tão presente na minha vida, é sempre uma ajuda nos momentos mais difíceis. Aos meus tios, sempre disponíveis para me ajudar, quer como meu ajudante nas minhas aventuras, quer como taxista ou ainda como costureira. Ao meu afilhado pelas brincadeiras e descontrações proporcionadas. Quero agradecer também à restante família pelo apoio prestado ao longo destes 5 anos e que não consigo enumerar aqui.

Aos meus orientadores, ao Bruno e ao professor Aníbal, pela ajuda concedida ao longo deste último trabalho realizado para esta dissertação. Em especial ao Bruno pelo tempo e conhecimentos disponibilizados sem os quais eu não conseguia estar aqui a escrever. Uma palavra para o restante pessoal do laboratório que me ajudou com o seu conhecimento pratico de algumas ferramentas disponíveis.

Foram vários anos que passei nesta casa, na qual fiz amizades para a vida. Amizades que nunca tinha pensado serem possíveis dada a diferença nas idades, amizades estas que me marcaram. Às grandes amizades com o pessoal que começou esta aventura em 012, às brincadeiras que fizemos, às dificuldades passadas em conjunto e à sede que passamos juntos. Somos muitos, podemos ter as nossas diferenças mas temos algo que sempre nos junta. Iremos continuar a ser unidos por muitos e por muitos anos. Às amizades mais recentes e sem as quais estes últimos anos não tinham sido os mesmos que foram. Todas esta amizades foram possibilitadas pela roupa usada e pela forma de viver a vida académica que nos faz crescer como pessoas e desenvolve capacidades que nunca pensamos vir a desenvolver.

A todos um obrigado!

José Tiago Meireles

“If the facts don’t fit the theory, change the facts.”

Albert Einstein

Conteúdo

1	Introdução	1
1.1	Contexto	1
1.2	Motivação	1
1.3	Objetivos	1
1.4	Estrutura da Dissertação	2
2	Revisão Bibliográfica	3
2.1	Localização subaquática	4
2.1.1	Sensores inerciais	5
2.1.2	<i>Transponders e beacons</i>	5
2.1.3	Geofísicos	7
2.1.4	Fusão dos dados	8
2.2	Formas de locomoção	9
2.2.1	Rodas	10
2.2.2	Lagartas	10
2.2.3	Pernas	11
2.2.4	Articulado	13
2.2.5	Híbridos	13
3	Definição do problema	15
3.1	Requisitos	15
3.2	Conceitos do sistema	16
4	Desenvolvimento	21
4.1	Projeto final	21
4.2	Escolha e caracterização de componentes	22
4.2.1	Atuação	23
4.2.2	Sensores	24
4.2.3	Controlo	29
4.2.4	Armazenamento de energia	32
4.2.5	Elementos mecânicos	33
4.3	Integração	34
5	Software	37
5.1	Nível inferior	37
5.2	Nível intermédio	38
5.2.1	PCB do laboratório	38
5.2.2	Arduino	39

5.3	Nível superior	40
5.3.1	<i>Driver</i>	40
5.3.2	Programa de controlo	40
5.3.3	Teste com os microprocessadores	41
5.3.4	Odometria	41
5.3.5	Função GotoXY	42
6	Testes e Resultados	45
6.1	Locomoção	45
6.2	Cálculo das constantes para a odometria	45
6.3	Localização	47
7	Conclusão	49
7.1	Cumprimento dos objetivos e requisitos	49
7.2	Trabalho futuro	49
	Referências	51

Lista de Figuras

2.1	Esquema dos métodos de navegação subaquática. Fonte: [1]	4
2.2	Os sistemas que utilizam <i>Transponders</i> . Fonte:[1]	6
2.3	Localização do AUV com uma única baliza. Fonte: [1]	6
2.4	Localização do AUV com um CNA. Fonte: [1]	7
2.5	Diferentes tipos de sonares. (a) <i>Sidescan</i> , (b) <i>Multibeam</i> , (c) <i>Forward Looking</i> , (d) <i>Mechanical Scannig and Imaging</i> e (e) <i>Synthetic Aperture</i> . Fonte: [1]	8
2.6	Estimação da posição. Fonte: [1]	8
2.7	Métodos de aplicação do algoritmo SLAM. (a) <i>Feature-based</i> , (b) <i>View-based</i> . Fonte: [1]	9
2.8	O robô CMOVE enquanto está a ser recuperado. Fonte: [2].	10
2.9	O robô TRAMPER em testes. Fonte: [3].	11
2.10	O robô CR200 com todos os seus equipamentos. Fonte: [4].	12
2.11	O robô AQUA a nadar. Fonte: [4].	13
2.12	O robô AZIMUT. Fonte: [5]	14
3.1	Conceito do sistema	16
3.2	As soluções idealizadas.	18
4.1	Vista detalhada dos subsistemas.	22
4.2	<i>System Breakdown Structure</i>	22
4.3	O subsistema da atuação.	23
4.4	O motor M200 da Bluerobotics. Fonte: [6]	24
4.5	As lagartas da Lynxmotion.	25
4.6	O subsistema da sensorização.	25
4.7	Sensor AM4096 na placa de avaliação RMK4. Fonte: [7]	26
4.8	<i>Setup</i> para os testes.	27
4.9	Os sinais A, B e Ri de um <i>encoder</i> incremental. t_{DT} corresponde à distância entre a transição entre os dois sinais e depende da resolução do <i>encoder</i> . Fonte: [8]	27
4.10	Sinais U, V e W.	28
4.11	Sinal U gerado pelo Arduino.	28
4.12	O subsistema de controlo.	29
4.13	O ESC de 60A da HobbyKing.	31
4.14	A PCB desenvolvida pelo laboratório.	32
4.15	A bateria utilizada.	33
4.16	Peças desenhadas em SolidWorks.	34
4.17	Diagrama do sistema final.	35
4.18	Vista em perspetiva do protótipo.	35
4.19	Vista frontal do protótipo.	36

4.20	Vista superior do protótipo.	36
5.1	O adaptador para ligar o ESC ao computador.	38
5.2	O diagrama de estados da PCB.	39
5.3	Sequência da função UVW do Arduino.	39
5.4	O nível superior de controlo.	40
5.5	O processo para obter a posição através da odometria.	41
5.6	Diagrama de estados da função gotoXY. Fonte:[9].	44
6.1	O deslocamento obtido em função do comando.	46
6.2	O percurso estimado e a posição real e estimada do robô.	47

Lista de Tabelas

2.1	Vantagens e desvantagens das rodas	11
2.2	Vantagens e desvantagens das lagartas	11
2.3	Vantagens e desvantagens das pernas	13
2.4	Vantagens e desvantagens dos robôs articulados	13
3.1	Os requisitos do sistema.	16
3.2	Tabela de classificação dos modos de locomoção. [10]	18
4.1	Características dos motores M200. Fonte: [6]	24
4.2	Características do sensor AM4096	26
4.3	Resultados do teste do <i>encoder</i> incremental	27
4.4	Características dos ESC	30
4.5	Resultados do teste do <i>encoder</i> incremental	31
5.1	Parâmetros para programação dos ESC.	37
6.1	Número de impulsos dos <i>encoder</i> com o deslocamento de 1 m.	46

Abreviaturas e Símbolos

AHRS	<i>Attitude and Heading Reference System</i>
AI	<i>Artificial Intelligence</i>
ASC	<i>Autonomous Surface Craft</i>
AUV	<i>Autonomous Underwater Vehicle</i>
CNA	<i>Communication and Navegation Aid</i>
CNC	<i>Computer Numeric Control</i>
DEEC	Departamento de Engenharia Eletrotécnica e de Computadores
DC	<i>Direct Current</i>
DVL	<i>Doppler Velocity Log</i>
EKF	<i>Extended Kalman Filter</i>
EMF	<i>Electromagnetic Field</i>
ESC	<i>Electronic Speed Controller</i>
GIB	<i>GPS Intelligent Buoy</i>
GPS	<i>Global Positioning System</i>
IMU	<i>Inertial Measurement Unit</i>
INS	<i>Inertial Navigation System</i>
LBL	<i>Long Baseline</i>
PC	<i>Personal Computer</i>
PCB	<i>Printed Circuit Board</i>
PWM	<i>Pulse-Width Modulation</i>
SBS	<i>System Breakdown Structure</i>
SEIF	<i>Sparse Extended Information Filter</i>
SHAD	<i>Small Hovering AUV with Differential actuation</i>
SLAM	<i>Simultaneous Localization And Mapping</i>
TDMA	<i>Time Division Multiple Access</i>
TOF	<i>Time Of Flight</i>
USBL	<i>UltraShort Baseline</i>

Capítulo 1

Introdução

1.1 Contexto

Esta dissertação é desenvolvida no contexto dos veículos submarinos autônomos e a sua localização relativa. A maioria das explorações e intervenções submarinas são realizadas com recurso a veículos cujos meios de locomoção se baseiam em propulsores. As vantagens relativas ao uso destes mecanismos são várias, mas podem tornar-se pouco adequadas quando o veículo está próximo do fundo ou até pousado no solo [11]. Com o veículo pousado no solo levanta-se uma série de problemas, como que tecnologia utilizar e ainda como implementar essas tecnologias, ao qual se pretende responder com este trabalho.

1.2 Motivação

Com o aumento da exploração subaquática, cresce a necessidade de desenvolver novas tecnologias para ajudar a descobrir o grande e desconhecido mundo subaquático.

Com o intuito de desenvolver um robô autônomo capaz de se deslocar em superfícies subaquáticas, pretende-se desenvolver uma tecnologia de exploração submarina que irá permitir que um veículo se movimente no solo com maior facilidade, de recolher amostras do solo e de seguir percursos que são impossíveis com propulsores. Outra aplicação deste tipo de robô é o suporte de outros que se desloquem perto deste, pois pode funcionar como referência para a localização dos mesmos.

1.3 Objetivos

Este trabalho tem como objetivo principal o desenvolvimento de um robô capaz de se deslocar em superfícies subaquáticas. Devido à natureza do trabalho, este tem vários objetivos que devem ser cumpridos:

- Projeto e desenvolvimento de um sistema de locomoção;

- Projeto e desenvolvimento de um sistema de localização;
- Integração do sistema de locomoção no módulo já desenvolvido;
- Integração de um sistema de localização relativa;
- Projeto de manobras elementares;

A nível pessoal este trabalho tem como objetivo o desenvolvimento de capacidades e conhecimentos fundamentais para projetos futuros.

1.4 Estrutura da Dissertação

Este documento está organizado de forma a descrever o trabalho desenvolvido, seguindo uma linha temporal que começou na unidade curricular de preparação da dissertação.

No Capítulo 2 descreve-se o estado da arte sobre veículos submarinos autónomos capazes de se deslocarem no solo e ainda sobre as tecnologias para a localização subaquática.

A definição do problema através da definição de requisitos e do conceito do sistema encontra-se no Capítulo 3.

No Capítulo 4 introduz-se a principal ideia para a solução final, é realizada a escolha dos diversos componentes e ainda a caracterização dos mesmos através da realização de alguns testes.

No Capítulo 5 é descrita a implementação realizada a nível de software.

Os resultados obtidos no contexto dos testes realizados ao sistema completo são apresentados no Capítulo 6.

No último capítulo, o Capítulo 7 são apresentadas as conclusões retiradas deste projeto assim como o trabalho futuro.

Capítulo 2

Revisão Bibliográfica

Este capítulo apresenta o levantamento ao estado da arte relativo ao problema da locomoção em superfícies subaquáticas de robôs autônomos e a sua localização. Com o problema em mente foi necessário definir algumas palavras-chaves ou conceitos-chaves fundamentais para poder executar uma pesquisa devidamente fundamentada e orientada para o problema em questão.

Em primeiro lugar a palavra *rover*, nome dado a um veículo de exploração espacial desenhado para movimentar-se na superfície de um planeta ou satélite. Os *rover* são desenhados para funcionarem em ambientes extremamente agressivos, como é também o fundo dos oceanos. Verificamos que existem vários tipos de *rovers*, com diversas características e funcionalidades que dependem da objetivo e da localização da missão [12]. Como o objetivo é ter um veículo capaz de se deslocar em superfícies subaquáticas, encontrou-se os *Crawler*, veículos submarinos desenhados para se deslocarem no fundo dos oceanos. Estes veículos têm características de um **ROV** dada a um veículo submarino operado remotamente, e ainda de um *rover*, pois têm formas pouco hidrodinâmicas, deslocam-se a baixa velocidade e ainda são capazes de realizarem manobras autônomas. Verificou-se que entre os *Crawler* encontrados, existiam diversos **modos de locomoção**. Foi então necessário uma pesquisa para melhor perceber as características, as vantagens e desvantagens dos diferentes modos de locomoção.

Por outro lado, existem os **AUV**, nome dado a um veículo submarino autônomo, onde se encontram diversas implementações e configurações. A diferença entre os AUV e os ROV estão como o nome indica no modo em que é comandado. Normalmente, os ROV são veículos maiores, de formas menos hidrodinâmicas e conseqüentemente operam a velocidades mais baixas. Por outro lado, os AUV são veículos tipicamente com a forma de torpedos, capazes de se deslocarem a tipicamente maiores velocidades pela forma mais hidrodinâmica [13].

Como o problema inicial passa também pela capacidade de localização do robô debaixo de água, foi realizada uma pesquisa sobre **localização subaquática**, onde se encontrou diversos métodos de localização subaquática, com as suas características e sensores necessários, na sua maioria sensores acústicos. Foi realizado um levantamento sobre os métodos de **localização relativa de robôs terrestres** baseada na informação de sensores no próprio robô. A **navegação de robôs**, quer em terra como na água, também foi alvo da pesquisa realizada. Navegação entende-se pela

capacidade de um robô se guiar de um ponto para o outro sem se perder enquanto que por localização entende-se pela capacidade de um robô se localizar num referencial ou mapa.

Com base nestas palavras-chaves ou conceitos-chaves, com ênfase para os **Crawler**, modos de locomoção terrestre e a localização subaquática foi realizado um levantamento bibliográfico direcionado para soluções tecnológicas relacionadas com o tema desta dissertação.

2.1 Localização subaquática

A localização subaquática continua a ser uma parte mais ou menos complexa e essencial no desenvolvimento de robôs submarinos. Devido ao meio, constituído por água, o principal sistema utilizado para localização à superfície de locais abertos e no ar, o *Global Positioning System* (GPS), assim como sinais de rádio frequência, são fortemente atenuados na água. Para ultrapassar estas limitações, as tecnologias mais usadas para localização subaquática incluem sistemas baseados em visão, sensores inerciais ou pela trilateração/triangulação através de sinais acústicos (figura 2.1) [1].

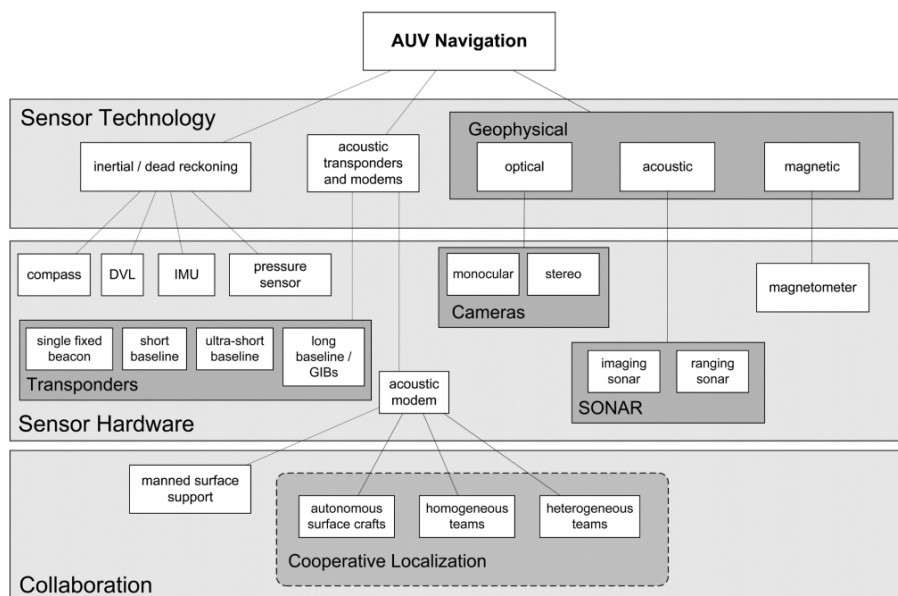


Figura 2.1: Esquema dos métodos de navegação subaquática. Fonte: [1]

Os sensores tipicamente usados para a localização de robôs submarinos incluem:

- Inerciais como *encoder*, acelerómetros e giroscópios, em que o erro de posição aumenta com o tempo.
- *Transponders* e modems acústicos baseados na medição do tempo de voo (TOF, do inglês *time of flight*).
- Geofísicos que utilizam referências externas do ambiente para navegação.

2.1.1 Sensores inerciais

Com base na informação dos sensores e sabendo a posição original, o robô consegue estimar a sua posição com base na velocidade e nas equações cinemáticas. A estimativa resultante deste método é que tem um erro associado devido sobretudo ao *offset* dos sensores. Sendo por isso necessário complementar usando outros métodos para estimação da posição, de forma direta como por exemplo o GPS, sinais acústicos ou sensores geofísicos.

2.1.1.1 Encoder

Dentro dos *encoder* existem dois grandes tipos: os incrementais e os absolutos. Os *encoder* incrementais têm a capacidade de informar sobre a velocidade e direção do movimento enquanto os sensores absolutos informam sobre a posição exata do eixo [14]. Os *encoder* podem ser magnéticos, indutivos, capacitivos ou óticos. Têm todos o mesmo princípio de funcionamento, apenas muda a forma como detetam movimento.

2.1.1.2 IMU

É uma unidade composta por uma combinação de sensores como acelerómetros, giroscópios e magnetómetros para medir as forças aplicadas no veículo, que posteriormente serão usadas para calcular a pose do mesmo [15].

2.1.1.3 DVL

Através da utilização de sensores de movimento Doppler para medir distâncias ao solo é possível determinar a velocidade do AUV. Com base na informação é possível calcular a pose do AUV. O sensor envia um conjunto de ondas sonoras e através do efeito de Doppler que verifica na reflexão das mesmas é possível calcular a distância ao solo.

2.1.2 Transponders e beacons

Os seguintes métodos têm como base os sensores acústicos. Funcionam no princípio da medição do TOF dos sinais acústicos.

2.1.2.1 USBL [16] e SBL [17]

No sistema *UltraShort Baseline* (Figura 2.2a) os *Transponders* estão localizados perto uns dos outros e a distância é calculada pela diferença no tempo de receção do sinal nos *Transponders* que normalmente estão instalados num navio à superfície. É recomendado para pequenas distâncias.

O sistema *Short Baseline* (Figura 2.2b) difere do USBL pelo facto que os *Transponders* estão mais distanciados uns dos outros. A distância é calculada pela triangulação do sinal. É limitado pela distância entre os *Transponders*, sendo que quanto maior, menor é o erro de precisão. É limitado ainda pela distância entre os *Transponders* e os recetores.

Nas Figuras 2.2a e 2.2b é apresentada uma configuração possível para os sistemas USBL e SBL. Neste caso os *Transponders* estão localizados no navio, no entanto podem estar localizados no próprio AUV.

2.1.2.2 LBL e GIB [15]

No sistema *long baseline* a localização é alcançada pela triangulação dos sinais acústicos provenientes de balizas instaladas no fundo do mar ou dos GIB, balizas que são colocadas à superfície na qual a sua posição é conhecida usando o sistema GPS. O principal obstáculo a este sistema é o custo e tempo associado à criação da rede.

Na Figura 2.2c é possível observar o sistema.

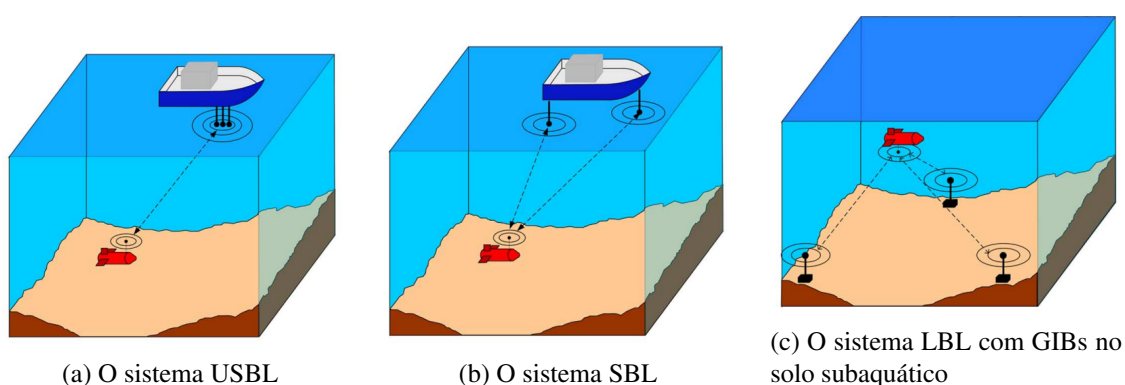


Figura 2.2: Os sistemas que utilizam *Transponders*. Fonte:[1]

2.1.2.3 Single fixed beacons

Em alternativa ao LBL (Figura 2.3), é possível a utilização de apenas uma baliza. Este sistema garante a melhor estimativa da posição do AUV quando, por exemplo, o AUV está a circular numa direção tangencial à localização da baliza.

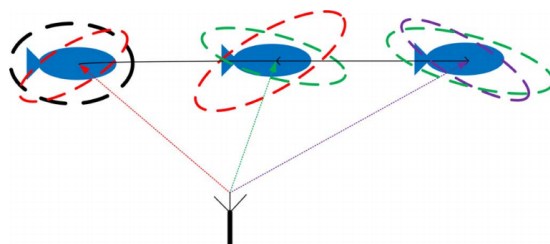


Figura 2.3: Localização do AUV com uma única baliza. Fonte: [1]

2.1.2.4 Modems acústicos

É também possível a comunicação de dados por ondas sonoras, em pequenos pacotes. Nestes pacotes é possível enviar a localização do navio ou veículo auxiliar e com base no TOF calcular

a distância entre os dois. Este sistema pode ser implementado de diversas formas, sendo que a diferença está no veículo auxiliar que podem ser navios tripulados, navios não tripulados (ASC) e equipas de AUV formadas por *Communication and Navigation Aid*, CNA. A principal vantagem destes em relação aos métodos anteriores é o facto de estes serem móveis, consequentemente conseguirem acompanhar o AUV e comunicarem a sua localização entre si reduzindo assim o erro de posição. Na Figura 2.4 é possível verificar a localização de um AUV com a utilização de um CNA.

As principais desvantagens das comunicações acústicas são:

- Largura de banda baixa utilizando o sistema de *Time Division Multiple Access* (TDMA).
- Taxa de dados transmitidos baixa.
- Latência elevada devido à velocidade de propagação, a velocidade do som.
- Velocidade do som depende de fatores externos como a temperatura e a salinidade da água.
- O sistema é muito incerto, com muitas perdas de dados.

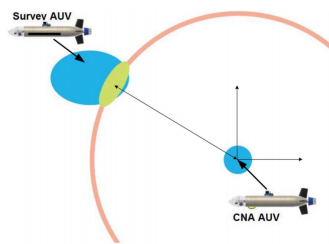


Figura 2.4: Localização do AUV com um CNA. Fonte: [1]

2.1.3 Geofísicos

Entende-se por sensores geofísicos sensores capazes de recolher características geofísicas do meio. Entre eles existem os sensores óticos e o sonar.

2.1.3.1 Óticos

A localização com base em sensores óticos passa pelo processamento de imagens para determinar a posição do robô. Necessita de pelo menos uma câmara monocular ou estéreo para capturar imagens. Este método tem pouco alcance devido tipicamente às características óticas do meio. Utilizado como fonte de informação nos algoritmos de localização mas pode também ser usado independentemente, por exemplo, na inspeção de tubos.

2.1.3.2 Sonar

A tecnologia sonar nos oceanos é mais antiga do que os AUV. Existem diversos tipos de sonares que são classificados dentro de duas grandes classes: sonar de imagem e de *ranging* sonar. Estes operam a frequências que dependem do alcance e resolução pretendidas. Em todos existem

sinais que são enviados e refletidos, sendo detetados pelos sensores e usados para criar um mapa, determinar medidas, características do solo e ainda obstáculos. Estes podem ser emitidos em várias direções, sendo esta uma característica que distingue os vários tipos de sonar.

Na Figura 2.5 é possível observar os diferentes tipos de sonares usados.

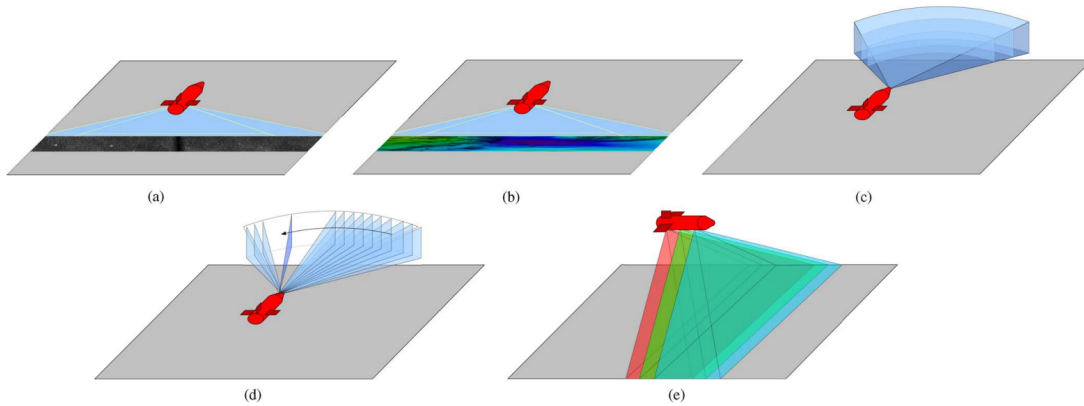


Figura 2.5: Diferentes tipos de sonares. (a) *Sidescan*, (b) *Multibeam*, (c) *Forward Looking*, (d) *Mechanical Scanning and Imaging* e (e) *Synthetic Aperture*. Fonte: [1]

2.1.4 Fusão dos dados

Existem diversos métodos para a fusão dos dados obtidos para a localização e navegação dos veículos. A estimação do estado é a base de qualquer algoritmo de navegação (Figura 2.6). Os métodos de estimação são normalmente aplicados com base em filtros, em particular, aplicações do filtro de Kalman.

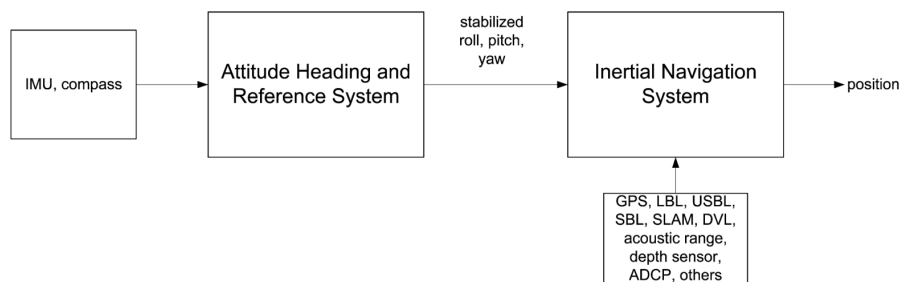


Figura 2.6: Estimação da posição. Fonte: [1]

2.1.4.1 Odometria

A odometria é um método que calcula a posição relativa do robô no solo com base na informação obtida no *encoder* ou de uma sequência de imagens. Com base na informação obtida, é calculado o deslocamento do robô e posteriormente atualizada a posição do robô. Este método é muito suscetível a erros sistemáticos pelo que necessita de correção e de uma referência independente para evitar que o erro aumente [14].

As equações utilizadas dependem da configuração do robô. As configuração mais utilizadas para robô com rodas são [9]:

- Tração diferencial, com pelo menos duas rodas com tração, fixas e paralelas.
- Tração síncrona, com pelo menos três rodas com tração e direção.
- Tração em triciclo, com duas rodas livres, fixas e paralelas e ainda uma roda com tração e direção, como um triciclo.
- Tração Ackerman, com duas rodas livres, fixas e paralelas e duas rodas com tração e direção, como um automóvel.
- Tração omnidirecional, em que cada roda pode movimentar-se em diferentes direções.

2.1.4.2 Simultaneous Localization and Mapping (SLAM)

Um exemplo de um algoritmo de localização e navegação é o SLAM [18]. Este é um processo de criação de um mapa do ambiente, posterior localização do robô no mesmo e de constante atualização do mapa e da posição do robô. A criação do mapa é feita pelo reconhecimento de características do meio ou por valores de medidas a pontos conhecidos no espaço, como é possível observar na Figura 2.7. Necessita de um grande poder de processamento e memória disponível.

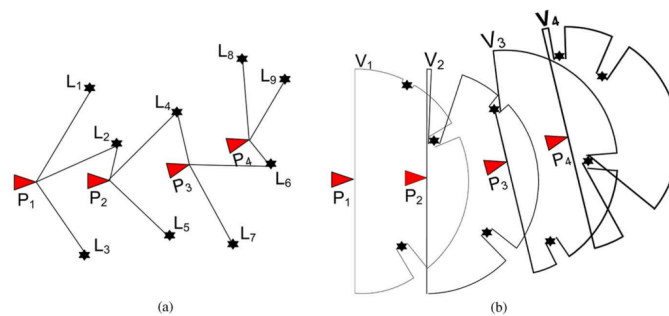


Figura 2.7: Métodos de aplicação do algoritmo SLAM. (a) *Feature-based*, (b) *View-based*. Fonte: [1]

2.2 Formas de locomoção

Existem várias soluções de locomoção em superfícies subaquáticas, entre as quais rodas, lagartas, pernas, a articulação do robô e os chamados híbridos, isto é, combinações de várias soluções de locomoção. Todas têm vantagens e desvantagens, dependendo de vários fatores como a tipologia e topologia do terreno.

O solo tem um papel determinante no deslocamento [19] e na escolha do tipo de solução a adotar. A sua composição, a sua inclinação, a sua movimentação e a saturação de água no mesmo são fatores que influenciam o desempenho do sistema de locomoção.

Outros fatores que influenciam o desempenho dos sistemas é a sua morfologia e o seu sistema de tração, sistemas estes que têm fatores associados como os graus de liberdade que têm, o diâmetro das rodas, a pressão que exerce e a razão de penetração no solo que será menor quanto maior for o número e a área dos contactos e ainda a folga que têm com o solo.

A maior parte deste fatores tem implicações com outros, como é o caso da penetração no solo que depende da pressão exercida no contacto com o solo e da saturação de água no mesmo, por exemplo, no caso do robô se deslocar em areia molhada a penetração no solo será maior do que se o deslocamento fosse efetuado em cimento.

Para melhor perceber as diferenças das várias formas de locomoção procedeu-se à recolha das suas principais características e as suas vantagens e desvantagens [10].

2.2.1 Rodas

Locomoção com rodas é a solução mais comum. Esta forma utiliza rodas para o contacto com o solo, onde quanto maior o número de rodas com tração, melhor o desempenho do sistema para se deslocar sobre as superfícies, o que providência um contacto mais estável, que pode ser melhorado com a integração de mais rodas. Esta forma permite também ter uma suspensão ativa e independente. Esta solução foi utilizada, por exemplo, no *Crawler CMOVES* (Figura 2.8). Este veículo subaquático desloca-se com rodas e é destinado a efetuar medições na superfície do solo [2]. Foi desenvolvido por uma parceria entre a universidade de Bremen/MARUM e a agência aeroespacial alemã. Este pode funcionar em modo autónomo com uma autonomia de 12 horas ou com uma ligação em fibra ótica. O sistema de locomoção é formado por quatro rodas independentes oferecendo ao robô qualquer direção de movimento. Para navegação e localização este robô usa o sistema DVL ao mesmo tempo que usa o sistema USBL. Na Tabela 4.2, pode observar-se as principais vantagens e desvantagens desta forma de locomoção.



Figura 2.8: O robô CMOVE enquanto está a ser recuperado. Fonte: [2].

2.2.2 Lagartas

Este método utiliza lagartas para efetuar o contacto com o solo. É considerada uma forma versátil e é capaz de ultrapassar grandes obstáculos e terrenos soltos como é exemplo a neve ou a areia. É muito usado em veículos militares e também em veículos de construção como

Tabela 2.1: Vantagens e desvantagens das rodas

Vantagens	Desvantagens
Rápido e suave em terrenos planos; Tecnologia muito robusta; Capacidade de cargas; Baixo consumo energético;	Não consegue ultrapassar obstáculos maiores que o raio da roda; Têm dificuldade em deslocar-se sobre buracos e contornos presentes no caminho;

retroescavadoras. O veículo subaquático TRAMPER (Figura 2.9) é um dos vários desenvolvido na aliança entre a HGF e a ROBEX [3]. Este é totalmente autónomo e tem como base o sistema de lagartas para deslocamento. O seu principal objetivo é a gravação da distribuição de oxigénio nos sedimentos durante um período de 1 ano. Tem dimensões de 1,5 metros de comprimento, 1,3 metros de largura e 1,2 metros de altura com um peso de 653 quilogramas e 20 quilogramas negativos de flutuação [20]. Na tabela 2.2 é possível observar algumas vantagens e desvantagens desta forma de locomoção.

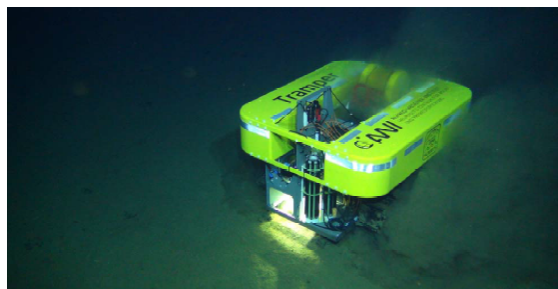


Figura 2.9: O robô TRAMPER em testes. Fonte: [3].

Tabela 2.2: Vantagens e desvantagens das lagartas

Vantagens	Desvantagens
Suave em terrenos planos; Tecnologia simples; Tração em solos soltos; Ultrapassa obstáculos e pequenos buracos; Capacidade de cargas pesadas;	Ineficiência devido ao atrito; Deslize em curvas; Agressivo no solo; Grande área de contacto com o solo; Sofre mais impactos devido à falta de suspensão;

2.2.3 Pernas

Devido à ineficiência das formas de locomoção anteriores e inspirado em animais com pernas capazes de ultrapassar qualquer tipo de terreno e obstáculo, uma alternativa às rodas e às lagartas são as pernas. Esta solução é inspirada na biologia e pode ter diferente número de pernas.

A sua posição afeta a estabilidade e a forma como se movimentam em relação ao corpo. As pernas podem rodar em torno de um eixo horizontal ou vertical. Na rotação em torno de um eixo

horizontal, como os humanos, as pernas estão em linha, viradas na mesma direção e debaixo do corpo. Na rotação em torno de um eixo vertical, mais semelhante a alguns insetos, as pernas saem do corpo para longe do centro de gravidade, o que faz com que sejam mais estáveis. Esta configuração é mais usada em robôs com mais de quatro pernas. O projeto das pernas também pode ser variado. Existem pernas articuladas com diferentes graus de liberdade ou articulações prismáticas. Na Tabela 2.3 listam-se as principais vantagens e desvantagens desta forma de locomoção.

Um exemplo de um veículo subaquático que utiliza pernas é robô CR200 (Figura 2.10). Foi desenvolvido pelo Departamento de Engenharia Oceanográfica da Universidade Nacional de Mokpo e o Departamento de Engenharia de Sistemas e Pesquisa do Instituto de Pesquisa e Desenvolvimento Oceânico da Coreia. Este tem como base um robô desenvolvido anteriormente, o *Crabster* [21]. Este robô é composto por seis pernas para se deslocar, sendo que as duas dianteiras também funcionam como braços manipuladores. Tem dimensões de 2,4 metros de comprimento, 1,25 metros de largura e no máximo 1,8 metros de altura, com um peso de 300 quilogramas em terra e 200 quilogramas quando submerso. Este é capaz de atuar em zonas pouco profundas onde existem correntes fortes devido a sensores que detetam estas correntes e na qual o robô assume uma posição de equilíbrio em relação às mesmas, como uma lagosta. O CR200 utiliza a tecnologia USBL assim como um sonar, uma câmara a cores e uma câmara acústica para a sua localização e navegação [22] [4].

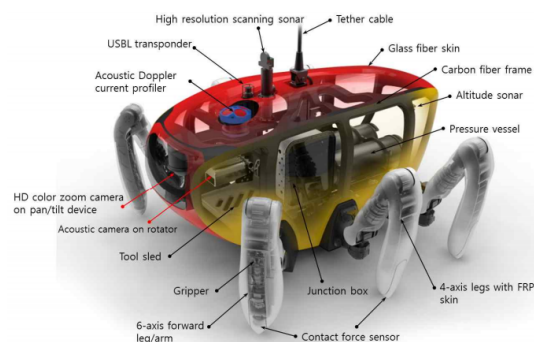


Figura 2.10: O robô CR200 com todos os seus equipamentos. Fonte: [4].

O robô AQUA (Figura 2.11) é outra possibilidade para a utilização de pernas para a deslocação do veículo. É considerado um anfíbio devido à sua capacidade de nadar e andar quer ao longo da costa como no fundo do mar. Este é baseado no *RHex* [23] e foi desenvolvido pelo laboratório de robótica ambulatória na Universidade McGill em colaboração com a Universidade do Michigan, a Universidade da Califórnia em Berkeley e a Universidade Carnegie Mellon. A capacidade de nadar vem das suas pernas que funcionam como remos [24], dando ao robô controlo sobre os seus movimentos exceto no deslocamento lateral, o *sway*. Tem dimensões de 0,65 metros de comprimento, 0,45 metros de largura e 0,13 metros de altura e desloca cerca de 16 quilogramas de água. Utiliza sensores inerciais e a visão estéreo em tempo real para aplicar o processo SLAM, assim como a tecnologia LBL com o apoio de GIBs [25].

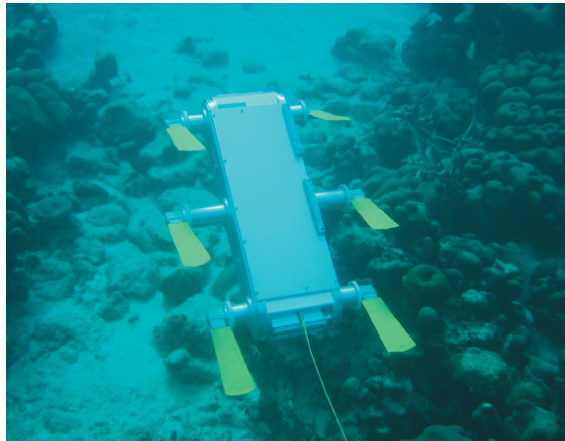


Figura 2.11: O robô AQUA a nadar. Fonte: [4].

Tabela 2.3: Vantagens e desvantagens das pernas

Vantagens	Desvantagens
Maior adaptabilidade ao terreno devido ao maior número de contactos; Contactos independentes; Suspensão ativa;	Composto por mais subsistemas; Maior peso de componentes; Baixa velocidade; Baixa capacidade de carga; Impactos com o solo a cada passo;

As pernas quando comparadas com as rodas conseguem um melhor controlo da força de reação do solo e do ângulo de contacto.

2.2.4 Articulado

Esta solução de locomoção é usada por sistemas que são compostos por diferentes segmentos ligados entre si. A locomoção é atingida pela variação coordenada dos segmentos, semelhante à locomoção de serpentes. Estes sistemas podem usar rodas ou lagartas para reduzir o atrito com o chão e obter uma maior eficiência. Na Tabela 2.4 são apresentadas algumas vantagens e desvantagens desta solução de locomoção.

Tabela 2.4: Vantagens e desvantagens dos robôs articulados

Vantagens	Desvantagens
Redundante; Adaptativo, navega entre os obstáculos; Consegue ultrapassar buracos;	Maior complexidade; Grande contacto com o solo;

2.2.5 Híbridos

Como o próprio nome indica, esta forma de locomoção é uma combinação de duas ou mais soluções. Esta combinação pode ter várias configurações, ao mesmo tempo ou altera conforme a

necessidade. Com esta implementação, o objetivo é obter as vantagens dos sistemas implementados. Como o número de combinações possíveis é grande, não é possível listar as vantagens e desvantagens desta solução, no entanto a complexidade da solução será sempre maior.

O robô AZIMUT (Figura 2.12) é um exemplo pois associa pernas com lagartas, sendo capaz de se deslocar no solo com alguma facilidade e ao mesmo tempo ter capacidade de ultrapassar obstáculos como escadas [5].



Figura 2.12: O robô AZIMUT. Fonte: [5]

Capítulo 3

Definição do problema

Com base na pesquisa bibliográfica, neste capítulo pretende-se definir os requisitos, as ideias de design para solucionar o problema e ainda definir, de forma abstrata, os principais subsistemas necessários para a solução final.

Os requisitos foram definidos em conjunto com o orientador durante uma reunião. Com base nos requisitos, foram idealizados vários conceitos para a solução, com as diferentes soluções para a locomoção. No entanto verificou-se a necessidade de um estudo mais completo dos mesmos, à luz de alguns aspetos fundamentais como a complexidade ou ainda a capacidade de adaptação ao tipo de terreno. Com base nesse estudo, os conceitos foram adaptados com vista aos modos de locomoção mais compatíveis com o problema.

Com as ideias do conceito definidas, uma arquitetura geral foi desenhada, começando por definir os subsistemas que teriam que existir, devido à necessidade de algumas características inerentes ao robô e ainda para cumprir os requisitos definidos, independentemente da solução final e comuns a todas as ideias.

3.1 Requisitos

Como em todos os projetos de sistemas de engenharia é necessário definir os requisitos para o sistema a desenvolver. Estes requisitos estão divididos em diferentes tipos, como por exemplos, os requisitos funcionais, os requisitos técnicos, os requisitos de desempenho ou ainda os requisitos de design. Em conjunto com o orientador foram definidos os requisitos presentes na Tabela 3.1.

Com o cumprimento destes requisitos, pretende-se obter uma solução capaz de se deslocar no fundo subaquático, a ser facilmente integrado no AUV SHAD existente no laboratório, habilitando-o a deslocar-se em meios diferentes dos quais ele foi inicialmente desenvolvido. O SHAD [26] distingue-se dos restantes AUV pela sua pequena dimensão com 0,12 metros de diâmetro, 0,9 metros de comprimento e 6,0 quilogramas de peso com uma forma cilíndrica tipo torpedo, pelo que o prototipo a desenvolver terá também de ser de dimensões reduzidas, peso reduzido e provocar o menor impacto hidrodinâmico possível. Deve ser capaz de operar a uma

Tabela 3.1: Os requisitos do sistema.

Requisito	Descrição	Tipo
1	Capacidade de deslocação no solo subaquático.	Funcional
2	Localização por odometria.	Funcional
3	Profundidade máxima de 100 metros.	Desempenho
4	Velocidade mínima de 10 centímetros por segundo.	Desempenho
5	Mínimo de 2 graus de liberdade.	Desempenho
6	Pequena dimensão.	Técnicos
7	Peso reduzido.	Técnicos
8	Compatibilidade com o AUV SHAD a nível elétrico e mecânico.	Design

profundidade máxima de 100 metros, com uma velocidade mínima de 10 centímetros por segundo, ter no mínimo dois graus de liberdade e capacidade de localização por odometria.

3.2 Conceitos do sistema

A solução será um sistema composto por vários subsistemas independentemente do design escolhido. Estes subsistemas serão depois implementados em conjunto de forma a obter um sistema capaz de cumprir com os requisitos impostos. Estes subsistemas incluem sistemas de controlo, de sensorização, de atuação, de armazenamento de energia e ainda um sistema responsável pela ligação entre a solução e o SHAD. O subsistema da atuação é necessário pois a solução necessita de se movimentar. O subsistema de sensorização é fundamental para que o sistema geral se consiga localizar. Ao subsistema de controlo é requerido o controlo dos subsistemas de atuação e sensorização. O subsistema de armazenamento de energia é responsável pelo fornecimento de energia para todos os restantes subsistemas. O subsistema de ligação entre a solução e o SHAD garante que ambos os sistemas podem funcionar como um só.

Na Figura 3.1 é possível observar o diagrama de blocos do sistema.

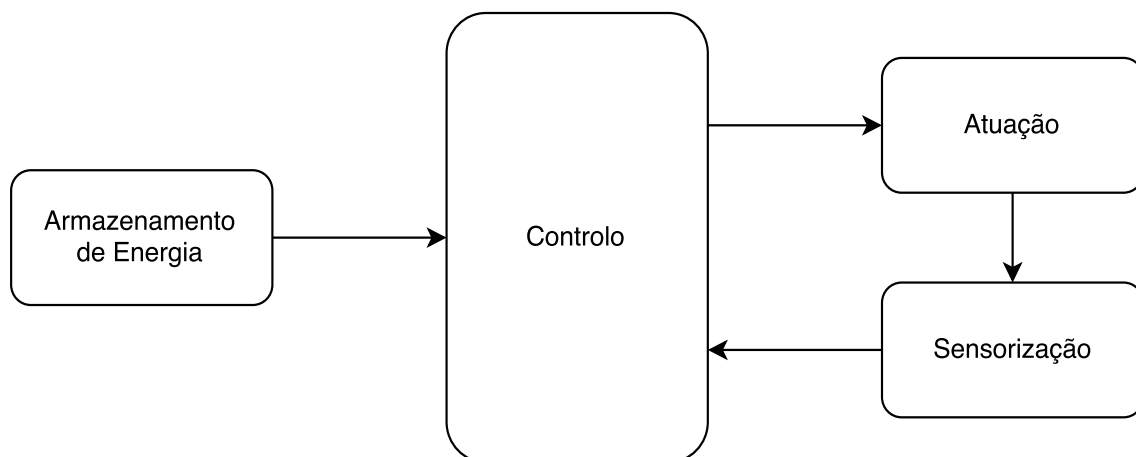


Figura 3.1: Conceito do sistema

Como é possível observar no diagrama de blocos, o sistema é dividido em quatro subsistemas. É possível também definir os objetivos para cada um dos subsistemas:

- O subsistema de armazenamento de energia deverá ser capaz de alimentar os restantes subsistemas.
- O subsistema de atuação deverá ser capaz de fazer o robô movimentar-se.
- O subsistema de sensorização deverá ser capaz de aferir o deslocamento efetuado pelo robô.
- O subsistema de controlo deverá ser capaz de controlar os subsistemas de atuação, sensorização, processar ordens recebidas e enviar informações para o utilizador.
- O subsistema para integração no SHAD, que deverá permitir uma fácil integração da solução, isto é, sem grandes alterações físicas e elétricas em ambas as partes, um sistema do tipo *plug-and-play*. Para tal, é necessário que ambos os sistemas funcionem na mesma base em termos elétricos e sejam compatíveis a nível de dimensões, características físicas e funcionamento mecânico.

Para a solução final pensou-se em diversas ideias de design, todas com base em rodas ou lagartas. Foram pensadas também soluções em que o deslocamento é no sentido do comprimento do SHAD, deslocamento frontal, e no sentido perpendicular ao comprimento do SHAD, deslocamento lateral.

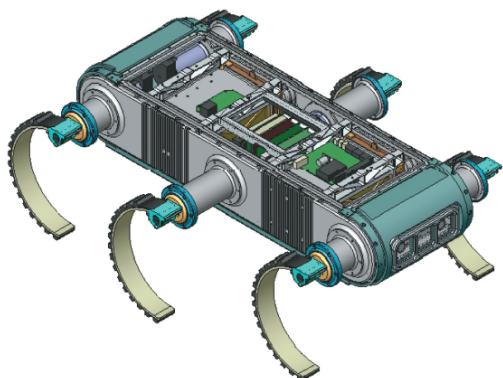
Ideias como colocar quatro patins no SHAD, com uma roda em cada, ou ainda umas pernas capazes de ficarem paralelas ao corpo do SHAD, oferecendo pouca resistência hidrodinâmica e quando o SHAD se aproxima do solo as pernas ficavam arqueadas, como no AQUA na Figura 3.2a.

No entanto, devido ao número de componentes necessários, foram desenvolvidas mais duas ideias de design que implicavam um menor número de componentes.

A primeira ideia consiste em ter uma roda em cada ponta do SHAD com múltiplos aros, com uma membrana a ligar os aros, a ser implementada diretamente no SHAD em que o deslocamento se efetuaria lateralmente como se fosse um *segway* visível na Figura 3.2b em que o corpo do AUV ficaria entre as rodas e os *endcaps* ficariam na parte de fora das rodas.

A segunda ideia consiste em ter um sistema de duas lagartas paralelas de pequena dimensão com deslocamento frontal. Este sistema seria independente do AUV, com todos os componentes necessários para o correto funcionamento integrados no próprio sistema, facilmente conectado física e eletricamente ao AUV para receber comandos e enviar informações. A forma física seria aproximadamente a mesma que o Trident da OpenROV [27] visível na Figura 3.2c em que as tiras pretas seriam substituídas por lagartas enquanto que o corpo central manteria a forma hidrodinâmica.

Em relação à componente mecânica da atuação, como verificado na revisão bibliográfica existem varias opções. Para melhor comparar os modos de locomoção, os mesmos foram comparados em diversos aspetos. Foi verificado o seu desempenho face ao tipo de terreno, a velocidade de



(a) O AQUA em modo de caminhar. Fonte [25]



(b) Um segway.



(c) O Trident da OpenROV. Fonte: [27].

Figura 3.2: As soluções idealizadas.

deslocamento possível, a sua agilidade, estabilidade e ainda a adaptabilidade ao tipo de terreno e ainda a possíveis obstáculos. A cada um dos critérios anteriores foram atribuídos pesos entre 5% e 20%. Foi ainda analisada a complexidade e dificuldade de implementação da solução, critérios estes com um peso maior pois são considerados mais importantes devido ao requisito de uma solução de dimensão reduzida.

Construiu-se a tabela 3.2 na qual se comparam 3 possíveis soluções: o primeiro sistema funciona com 2 rodas, o segundo sistema funciona com 2 lagartas e o terceiro funciona com 6 pernas [10].

Tabela 3.2: Tabela de classificação dos modos de locomoção. [10]

	Valor (%)	Rodas	Lagartas	Pernas
Terreno solto	20%	0	+1	0
Terreno Rugoso	5%	0	-1	+1
Velocidade	10%	+1	0	-1
Agilidade	10%	+1	0	+1
Estabilidade	10%	-1	+1	+1
Adaptabilidade	10%	0	0	+1
Complexidade	20%	+1	+1	-1
Implementação	15%	+1	+1	-1
Resultado	100%	4,5	6	-1

Como é possível observar na Tabela 3.2, a solução com melhor desempenho em terrenos soltos são as lagartas devido à grande superfície de contacto com o solo, o que permite também uma

maior estabilidade. No entanto em terrenos rugosos as pernas têm melhor desempenho devido ao maior número de pontos de contacto com o solo e menor área dos mesmos. Em termos de velocidade a melhor solução são as rodas, seguidas pelas lagartas e posteriormente as pernas, devido à necessidade de sincronizar os movimentos de vários componentes. Em termos de complexidade e implementação, as rodas e as lagartas exigem um menor número de atuadores e conseqüentemente menos controlo em relação às pernas, que necessitam de pelo menos um atuador por perna e em constante funcionamento para manter a posição pretendida.

Com base na informação recolhida a melhor solução são as lagartas, seguidos pelas rodas. Em último ficam as pernas devido essencialmente à complexidade e dificuldade de implementação do sistema. Esta dificuldade advém da necessidade de pelo menos um atuador por perna e a necessidade de 6 pernas para obter um veículo estável. Razão pela qual se descartou este sistema para a solução final.

Capítulo 4

Desenvolvimento

Com base nas ideias de design, verificou-se que independentemente do modo de locomoção escolhido, os subsistemas de sensorização e componente elétrica da atuação teriam sempre que fazer parte do sistema final. Verificou-se ainda que seriam compostos pelos mesmos componentes. Razão pela qual foi primeiro desenvolvido o trabalho tendo por objetivo selecionar os componentes de ambos os subsistemas. Com esses subsistemas definidos passou-se a escolher os componentes para a mecânica da atuação, escolhendo o modo de locomoção e conseqüentemente a ideia final para o sistema a desenvolver, com base nas ideias expostas no capítulo anterior. Ao mesmo tempo que os componentes iam sendo definidos, e os mesmos eram disponibilizados eram implementados e testados separadamente e posteriormente incluídos no sistema final.

4.1 Projeto final

O projeto final foi baseado na segunda ideia definida no Capítulo 3. Com a solução definida, foi preciso detalhar os subsistemas definidos anteriormente pois não representavam completamente o sistema pretendido.

Foi então necessário dividir o sistema em novos subsistemas (Figura 4.1). Entre os novos subsistemas destaque para os diferentes níveis de controle. A divisão do subsistema de controle em diversos níveis advém dos componentes disponíveis e ainda do controle necessário para poder implementar funções como a odometria e as manobras autônomas que o robô terá que realizar.

A arquitetura funcional do sistema é composta por seis grandes subsistemas, com alguns desses subsistemas divididos ainda para a melhor representação do subsistema. Foi criado o *System Breakdown Structure* (Figura 4.2) do sistema para uma melhor compreensão da arquitetura do sistema.

O subsistema da atuação foi dividido na sua componente elétrica e mecânica. Esta divisão advém da necessidade de implementar motores elétricos e ainda de componentes capazes de aplicar o trabalho realizado pelos motores em deslocamento físico.

O subsistema de controle foi dividido em três níveis, para obter um controle mais correto dos diversos componentes, como é o caso dos motores e sensores. É ainda necessário um controle

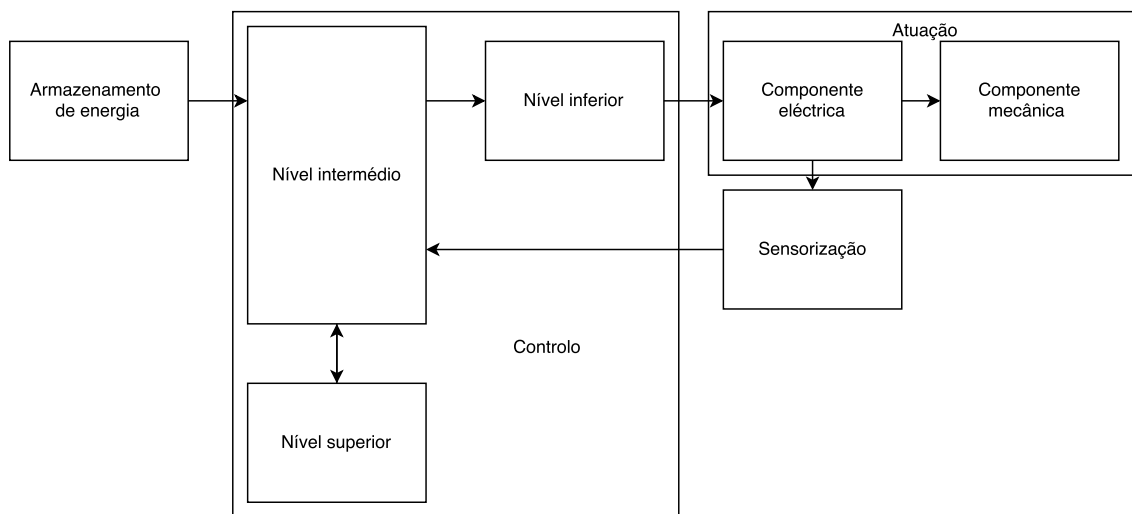


Figura 4.1: Vista detalhada dos subsistemas.

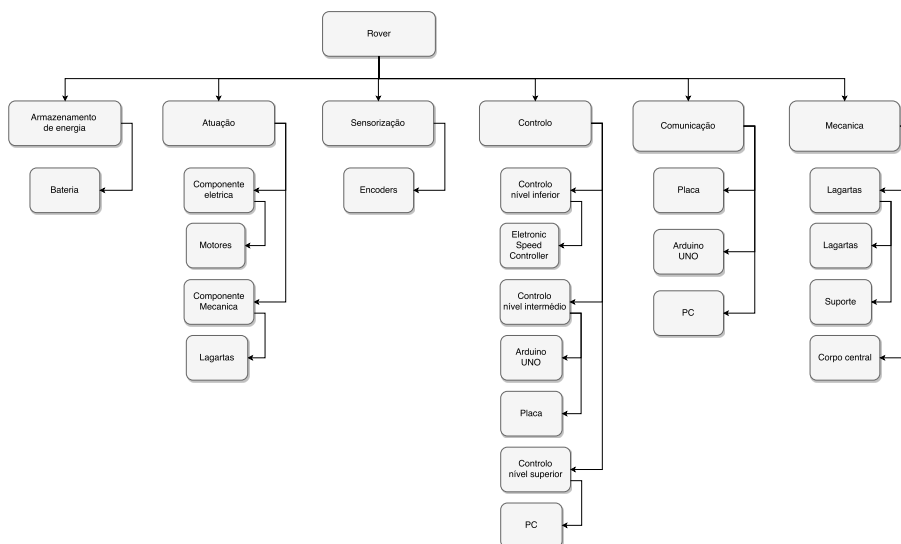


Figura 4.2: System Breakdown Structure

de alto nível para que o sistema possa realizar as funções que necessitam de um maior poder computacional para as quais foi projetado, como a execução de movimentos autónomos.

Com a divisão do subsistema de controlo em diferentes níveis é necessário que exista comunicação entre os mesmos para garantir que o sistema funciona como um só.

4.2 Escolha e caracterização de componentes

Para a solução final serão necessários elementos para componente eléctrica e mecânica da atuação, sensorização, controlo, armazenamento de energia e ainda elementos estruturais para suporte físicos dos diversos componentes.

Foram realizados levantamentos das tecnologias disponíveis no mercado e foram comparadas para permitir uma seleção fundamentada dos mesmos. Os componentes foram selecionados com base nas características, como a dimensão ou a composição, dando prioridade aos componentes que se enquadravam com a solução a nível físico e apresentavam resistência ao ambiente subaquático.

4.2.1 Atuação

Como já foi referido, este subsistema está dividido em duas componentes, a componente elétrica e a componente mecânica. A componente elétrica será centrada na escolha dos motores enquanto que a componente mecânica será centrada na escolha dos componentes de contacto com o solo.

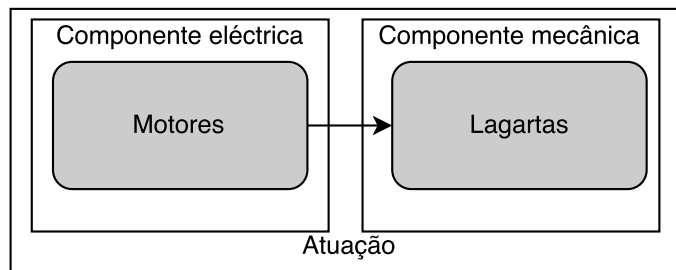


Figura 4.3: O subsistema da atuação.

4.2.1.1 Componente elétrica

Para a componente elétrica da atuação (Figura 4.3), devido aos requisitos do sistema, os motores têm que ser capazes de funcionar em ambientes subaquáticos a uma profundidade de 100 metros. Foi realizado um levantamento das tecnologias disponíveis para motores CC. Os motores CC capazes de movimentar o robô são os motores *brushless* CC, os motores *gimbal* e ainda os motores CC com escovas. No caso dos motores CC com escovas e motores *gimbal* seria necessário uma adaptação para que os mesmos possam funcionar num ambiente subaquático. Esta adaptação é no caso dos motores *gimbal*, o isolamento dos enrolamentos com resina epóxi ou no caso dos motores CC com escovas a instalação dos mesmos num recipiente capaz de isolar o motor da água. No caso dos motores *brushless* CC, existem no mercado motores já preparados para funcionarem em ambientes subaquáticos, estes motores são normalmente utilizados nos propulsores de AUV e ROV.

Com a diferença de preço entre os diferentes tipos de motores e o trabalho necessário para a adaptação dos mesmos, por uma questão de segurança, pois a adaptação poderia danificar os motores ou ainda reduzir o tempo de vida útil, optou-se pelos motores da Bluerobotics, os M200 (Figura 4.4).

Estes motores da Bluerobotics são máquinas síncronas de ímanes permanentes CC, vulgarmente chamados de motores *brushless* CC, com a característica de serem motores CC com três



Figura 4.4: O motor M200 da Bluerobotics. Fonte: [6]

fases [28]. A rotação acontece com a comutação das fases ativas, que gera um campo magnético girante que interage com o campo magnético criado pelos ímanes permanentes. Isto implica a necessidade de um controlador capaz de efetuar os comandos de comutação. Estes motores estão preparados para funcionar debaixo de água, pois têm os componentes isolados por plástico e resina epóxi. Na Tabela 4.1 é possível observar as principais características do motor M200.

Tabela 4.1: Características dos motores M200. Fonte: [6]

Tensão de operação	6-12 V
Corrente máxima	22 A
Potência máxima	350 W
Número de pares de polos	14

4.2.1.2 Componente mecânica

Para a componente mecânica da atuação (Figura 4.3), com base na ideia para a solução final foram escolhidas lagartas. Foi realizado um levantamento das opções disponíveis no mercado. As mesmas têm de ser capazes de resistir às condições subaquáticas e, tendo também em conta o fator de peso, as lagartas têm que ser de materiais leves e resistentes à água como o plástico ou a borracha, sem que existissem partes de metais suscetíveis a ficarem oxidados pela água. Foram encontradas diversas opções para a escolha das lagartas com base nas especificações fornecidas, no entanto devido ao preço, foi encontrado um sistema de lagartas modular o que permite um comprimento ajustável conforme a necessidade. Estas lagartas são produzidas pela Lynxmotion e tem duas polegadas de largura (Figura 4.5). Associadas a estas lagartas existem uma grande variedade de componentes fundamentais para o funcionamento correto das mesmas. Estes componentes incluem, por exemplo, as rodas dentadas sobre a qual as lagartas operam.

4.2.2 Sensores

No subsistema da sensorização (Figura 4.6) é necessário implementar sensores capazes de fornecer as informações necessárias para o resto do sistema como a posição e a velocidade do



Figura 4.5: As lagartas da Lynxmotion.

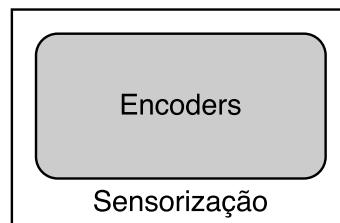


Figura 4.6: O subsistema da sensorização.

motor. Como um dos requisitos para o sistema é a capacidade de o robô se localizar através da odometria, é necessário um sensor *encoder* incremental em cada motor de cada uma das lagartas. Este sensor irá depois enviar dois sinais, que serão detetados por um circuito implementado em Arduino, capaz de contar o número dos impulsos fundamentais para a odometria. Existe ainda a necessidade de saber a posição exata do motor para o controlo dos motores *brushless* CC. Esta necessidade advém do facto de que o controlador dos motores precisa de saber a posição do motor para poder comutar as fases no momento correto.

Inicialmente foi realizado um levantamento dos sensores *encoder* disponíveis no mercado, assim como os tipos que existem. Existem vários tipos de *encoder* entre os quais os *encoder* óticos, magnéticos e ainda mecânicos. Como a solução necessita de funcionar em ambiente subaquático, a utilização de *encoder* magnéticos que não implicam contacto físico entre o sensor e o motor é a melhor opção. Dentro dos *encoder* magnéticos, existem ainda diferentes tipos, com diferença na posição onde é aplicado o elemento magnetizado, pode ser um íman no eixo do motor ou um anel magnético no exterior do motor. Devido fundamentalmente à necessidade de, independentemente da escolha, isolar o sensor com resina epóxi foi decidido escolher entre os sensores para soldar numa PCB que seria posteriormente desenvolvida. Na marca RLS verificou-se a existência de várias opções dentro dos *encoder* incrementais, e particularmente a capacidade de vários *encoder* disponibilizarem as saídas sobre a posição do motor, necessárias para o controlo dos motores. Foi então escolhido o sensor AM4096 [8] pois tinha todas as saídas necessárias para ambas as partes e ainda a capacidade de ser programado, alterando as definições do mesmo.

Na Tabela 4.2 é possível observar as características do sensor AM4096 e da PCB, designada por RMK4, disponibilizada pela RLS, visível na Figura 4.7.

Tabela 4.2: Características do sensor AM4096

Tensão de operação	3 V ou 5 V
Velocidade máxima	60000 rpm
Resolução máxima	12 bits
Saídas disponíveis	Incremental Serial SSI TWI Comutação UVW Linear Tacho Análogica
EEPROM programável	



Figura 4.7: Sensor AM4096 na placa de avaliação RMK4. Fonte: [7]

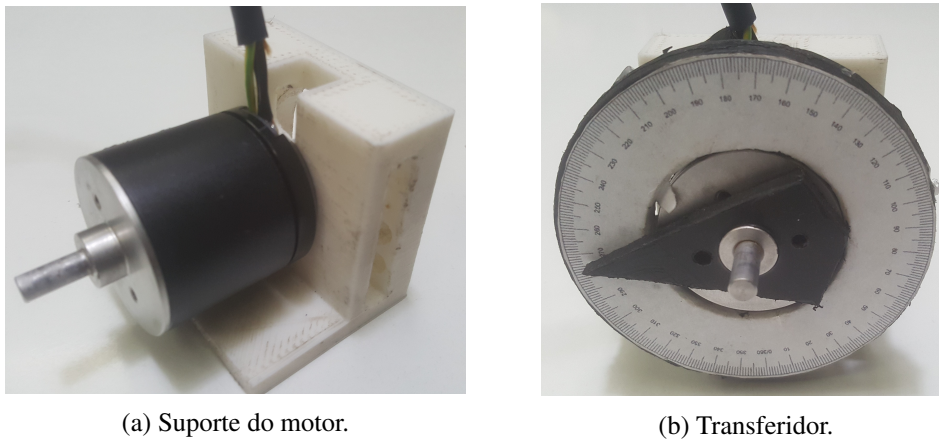
• Caracterização

Como o *encoder* tem que desempenhar várias funções no sistema, cada uma foi testada individualmente. Para a realização dos testes foi necessário adaptar os suportes para o motor e para o *encoder* para que os mesmos pudessem ser utilizados em cima de uma mesa, o que obrigou a imprimir um novo suporte para o motor e um suporte do sensor compatível. Foi também desenvolvido um transferidor e um apontador para poder medir a variação angular do motor, este foi cortado numa placa de PCB. Na Figura 4.8 é possível observar as peças criadas para a realização dos testes.

○ Encoder incremental

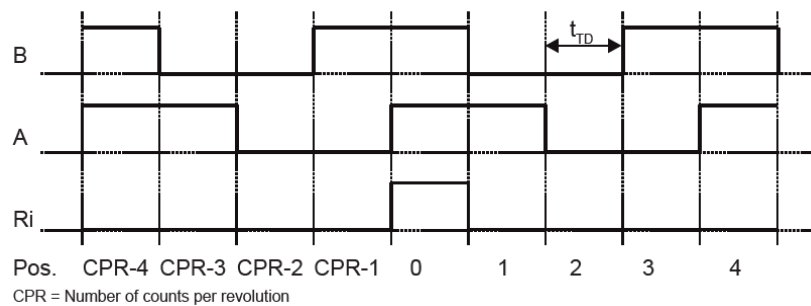
Este teste tinha como objetivo determinar a relação entre o número de impulsos do *encoder* com a rotação do motor. Para uma rotação de 360 graus o valor esperado é 2048 impulsos, devido à resolução do *encoder* de 12 bits por rotação [8], e a gama de valores ser $2^{n-1} = 2^{12-1} = 2048$, em que n é o número de bits.

Na Figura 4.9 é possível observar os sinais de saída num *encoder* incremental.



(a) Suporte do motor.

(b) Transferidor.

Figura 4.8: *Setup* para os testes.Figura 4.9: Os sinais A, B e Ri de um *encoder* incremental. t_{TD} corresponde à distância entre a transição entre os dois sinais e depende da resolução do *encoder*. Fonte: [8]

O motor foi rodado manualmente entre 0 e 360 graus, com incrementos de aproximadamente 90 graus. O sensor foi ligado a um Arduino para poder contar o número de impulsos gerados. Os valores obtidos estão representados na Tabela 4.3.

Tabela 4.3: Resultados do teste do *encoder* incremental

Ângulo (°)	número de impulsos
0	0
90	512
180	1047
270	1515
360	2050

Como é possível observar, os valores obtidos estão perto dos valores esperados, devido à falta de exatidão do transferidor e do facto de rodar o motor manualmente. É possível então assumir na odometria que uma rotação do motor corresponde a 2048 impulsos. Este valor irá ser usado posteriormente para calcular a relação entre o número de impulsos e o deslocamento de cada lagarta.

○ *Encoder UVW*

Este teste tinha como objetivo observar os sinais U, V e W [8]. Para obter os sinais é necessário alterar os registos do *encoder* em função do número de pares de polos do motor, que neste caso são 7. Os sinais esperados estão representados na Imagem 4.10 em que P é igual a 51,4 graus.

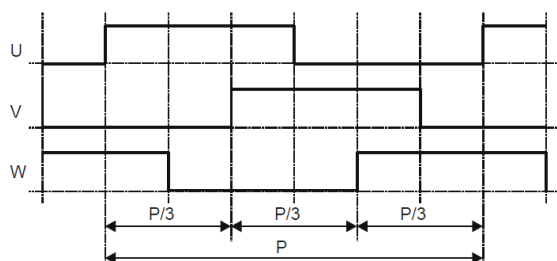


Figura 4.10: Sinais U, V e W.

Este sinal é esperado durante 51,4 graus, o que corresponde a 7 períodos elétricos numa rotação.

Devido à impossibilidade de reprogramar o sensor, foi necessário recorrer a outros sinais disponíveis do *encoder*, entre os quais um sinal linear proporcional à posição do motor, *Vout* [8], foi desenvolvido um programa para Arduino, que em função do valor da tensão *Vout*, gera os sinais U, V e W. Na Imagem 4.11 é possível observar o sinal U gerado numa rotação manual do motor com os 7 períodos dos sinais.

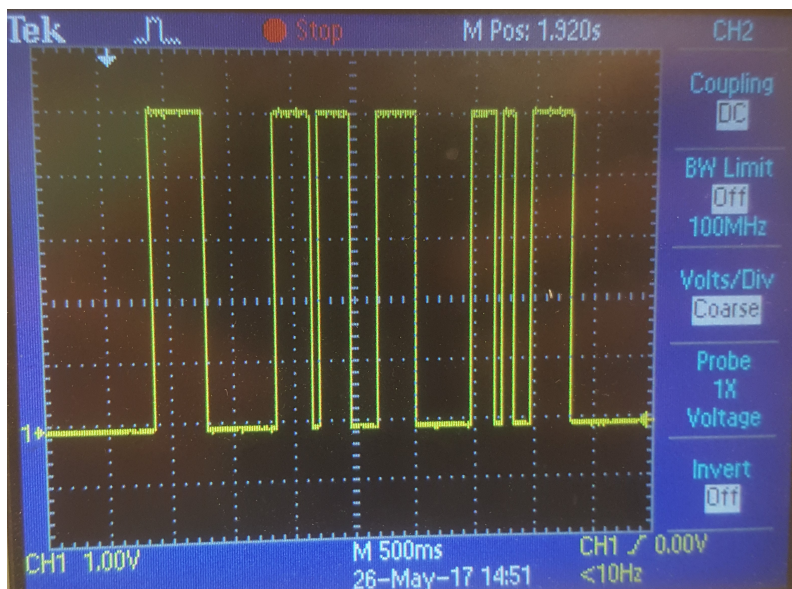


Figura 4.11: Sinal U gerado pelo Arduino.

Com base nos resultados obtidos optou-se por utilizar o Arduino para gerar os sinais U, V e W porque o código desenvolvido permitia alinhar os sinais com o posicionamento do motor através do incremento do valor do ângulo calculado, enquanto que os sinais diretos do sensor era necessário ajustar o posicionamento do íman.

4.2.3 Controlo

Como foi possível verificar anteriormente, o sistema necessita de vários níveis de controlo (Figura 4.12), cada um com as suas funções e dependência dos níveis mais próximos, o que implica que exista comunicação entre os mesmos. É possível definir 3 níveis de controlo, cada um com as suas funções.

- Nível inferior: corresponde ao controlo dos motores.
- Nível intermédio: responsável pelo controlo dos sensores e gerar os sinais necessários para o nível inferior.
- Nível superior: processamento de ordens e informação do estado do robô.

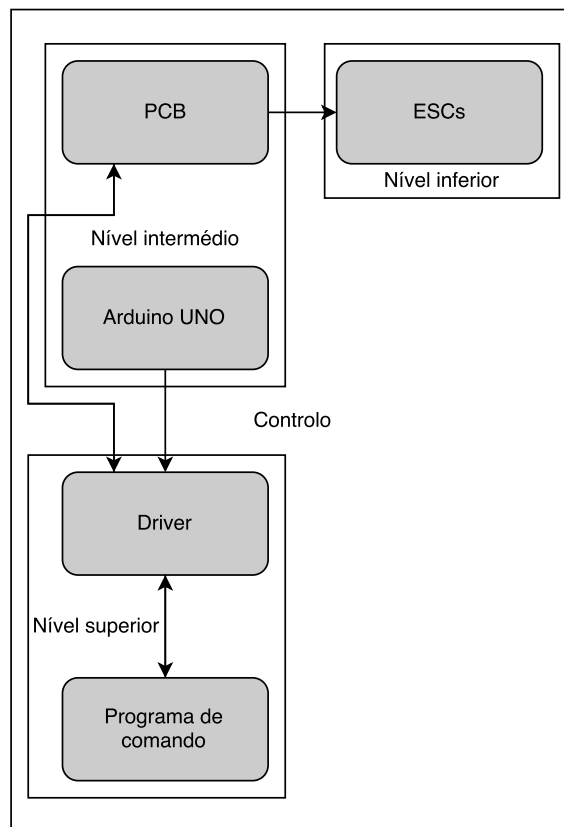


Figura 4.12: O subsistema de controlo.

Os constituintes de cada nível de controlo serão diferentes. A comunicação entre os diversos níveis será realizada conforme a informação que é pretendida comunicar.

Entre o nível inferior e o nível intermédio, é enviado um sinal de comando no sentido hierárquico, sob a forma de um sinal PWM correspondente à velocidade pretendida nos motores.

Entre o nível intermédio e o nível superior, a comunicação é efetuada nos dois sentidos, através de porta série, pois é necessário enviar ordens de velocidade dos motores no sentido hierárquico e ainda a informação dos sensores no sentido inverso.

4.2.3.1 Controladores de nível inferior

Estes controladores têm como objetivo o controlo dos motores *brushless* CC. O circuito mais básico possível de realizar este controlo é constituído por seis semi-condutores [28]. Existem no mercado diferentes circuitos em diversos níveis de implementação, com diferentes semi-condutores, com ou sem o circuito lógico para gerar os comandos de comutação e ainda com diferentes metodologias de controlo.

Pela necessidade de um correto controlo dos motores a baixas velocidades é necessário que o controlador tenha a posição atual do motor, fornecida pelo *encoder*. Com base na posição do motor, o controlador irá atuar nas fases correspondentes. Devido à grande oferta que existe em controladores de motores *brushless* CC, optou-se por utilizar controladores disponíveis comercialmente. No mercado existe uma grande oferta de controladores que têm como referência a velocidade pretendida, os ESC, divididos entre os *sensorless* ESC e os *sensored* ESC. Como o nome indica, a diferença entre os mesmos está na presença da informação dos sensores sobre a posição do motor. No caso dos *sensorless* ESC, a posição do motor é estimada através de *back-EMF* [29]. Este método é adequado para o controlo a altas velocidades mas no entanto, para velocidades mais reduzidas o seu comportamento não é satisfatório, devido ao sinal necessário para o controlo por *back-EMF* ser demasiado fraco. Neste caso, como é pretendido um controlo a baixas velocidades é necessário um ESC capaz de operar em função da informação do sensor. Foram escolhidos dois *sensored* ESC 60A da HobbyKing (Figura 4.13) com as suas características presentes na Tabela 4.4.

Tabela 4.4: Características dos ESC

Corrente nominal	60 A
Corrente máxima	240 A
Saída BEC	5.7 V / 3 A
Funcionamento programável via USB	

• Caracterização dos ESC

Para o funcionamento destes ESC, é necessário alimentá-los diretamente da fonte de energia do sistema, ligar as fases do motor às fases correspondentes do ESC, os sinais da posição do motor e ainda um sinal PWM, sinal este que varia com a velocidade pretendida. Este ESC tem ainda a particularidade de funcionar também no modo *sensorless* quando os sinais da posição do motor não são detetados.

Inicialmente foi pensada a utilização de *sensored* ESC, no entanto utilizou-se um *sensorless* ESC para testar as ligações do motor e verificar a velocidade mínima a que o motor rodava.

Foi por isso necessário verificar qual a velocidade mínima que o *sensorless* ESC consegue imprimir no motor. Utilizando um Arduino para gerar os sinais PWM necessários para o ESC, foram dados vários comandos e a velocidade de rotação do motor foi lida usando um tacómetro digital. Na Tabela 4.5 é possível observar a velocidade correspondente ao comando enviado. Para este teste, o ponto neutro é 1500 milissegundos, a velocidade máxima no sentido positivo

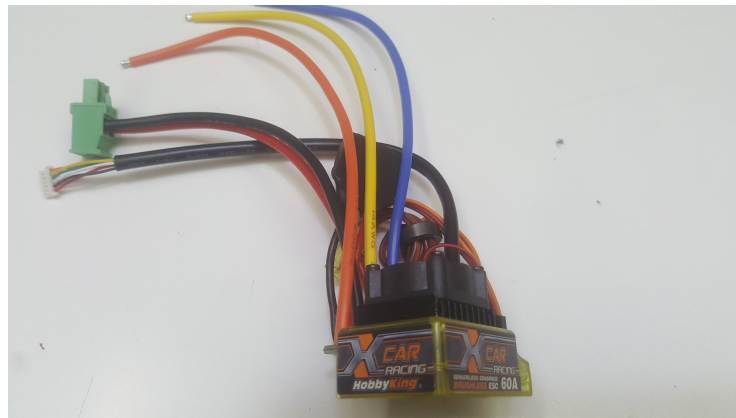


Figura 4.13: O ESC de 60A da HobbyKing.

é obtida com 1800 milissegundos e a velocidade máxima no sentido inverso é obtida com 1200 milissegundos. De referir ainda que o ESC utilizado tem uma redução por defeito na velocidade máxima no sentido inverso. Na Tabela 4.5 é possível observar os resultados obtidos.

Tabela 4.5: Resultados do teste do *encoder* incremental

Comando (ms)	Velocidade (rpm)	Comando (ms)	Velocidade (rpm)
1200	-1800	1500	0
1250	-1815	1545	185
1300	-1800	1550	275
1350	-1540	1560	540
1400	-1140	1570	840
1410	-900	1580	1100
1420	-750	1590	1400
1430	-430	1600	1720
1440	-230	1650	3040
1450	0	1700	3910
1455	0	1750	4440
1500	0	1800	4850

Apesar do ESC conseguir atingir velocidades baixas em ambos os sentidos, o motor aparenta não ter o binário máximo, pois era possível imobilizá-lo com a mão. Isto acontece porque o ESC está a estimar a posição do motor através de *back-EMF* geradas pelos enrolamentos desligados e quando a velocidade é baixa, a amplitude do sinal também é baixa e consequentemente não é detetada pelo ESC.

4.2.3.2 Controladores de nível intermédio

A este nível, é necessário gerar os sinais PWM para os ESC e processar os sinais dos *encoder* para obter a informação sobre a posição e velocidade do motor, necessárias para os restantes níveis de controlo.

Em relação aos sinais PWM, estes precisam de ser criados através de um circuito lógico baseado num microprocessador. Para esta função existe uma PCB já desenvolvida pelo laboratório, com base no microprocessador ATmega32u4. Esta PCB tem a particularidade de controlar e possibilitar a ligação de 2 ESC a uma única fonte de energia, analisando a corrente fornecida a cada um dos ESC. Esta PCB está configurada para receber comandos através de porta série, o que implica a existência de um nível de controlo superior para que seja possível a criação dos comandos.

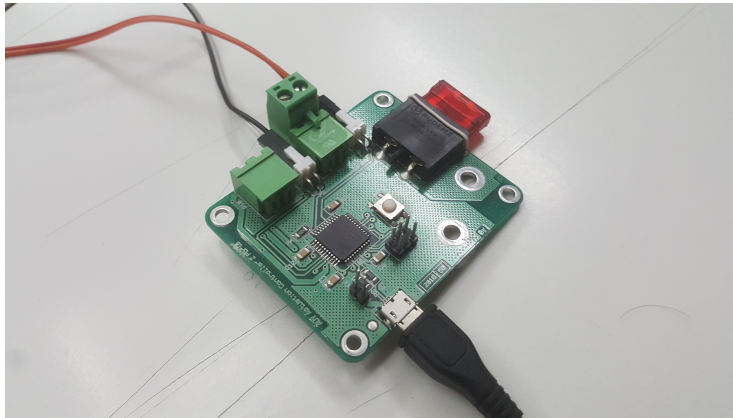


Figura 4.14: A PCB desenvolvida pelo laboratório.

Ainda a este nível, é necessário processar os sinais dos *encoder* para gerar os sinais da posição do motor, os sinais U V W, fundamentais para o funcionamento dos ESC a baixa velocidade e ainda a contagem de impulsos dos sinais A e B, sinais estes que são convertidos em distâncias percorridas por cada roda para depois serem usadas para implementar a odometria no nível superior. É utilizada a comunicação por porta série para enviar as distâncias para o nível superior de controlo.

Devido à falta de acesso às entradas na PCB anterior e ao tempo necessário para desenvolver uma nova, foi necessário recorrer a um Arduino UNO para implementar o processamento dos *encoder* e assim conseguir cumprir todas as funções a que este nível de controlo é proposto.

4.2.3.3 Controlador de nível superior

O nível superior de controlo é responsável por implementar as funções da odometria, de movimentos e manobras. Estas funções representam o nível mais elevado de controlo necessário para o funcionamento do robô. Funciona com base num PC Linux, pois a versatilidade, a possibilidade de servir de base a outros desenvolvimentos e ainda a possibilidade de ligar vários componentes por portas série diferentes.

4.2.4 Armazenamento de energia

Para alimentar os diversos sistemas necessários para o funcionamento do robô é necessário uma fonte de energia portátil, uma bateria. As baterias podem ser constituídas de diversos elementos, de diferentes capacidades e diferentes taxas de descarga.

Pela disponibilidade no laboratório e a compatibilidade com os restantes componentes, a bateria utilizada é uma bateria de polímeros de lítio com três células, 11,1 V e capacidade de 3700 mAh com uma taxa de descarga constante de 25C.



Figura 4.15: A bateria utilizada.

4.2.5 Elementos mecânicos

Os restantes componentes mecânicos para além dos componentes de atuação foram desenhados em SolidWorks.

Componentes como os suportes para os motores (Figura 4.16a), os suportes para os *encoder* (Figura 4.16b) e ainda os suportes para os ímanes dos *encoder* (Figura 4.16c) nos eixos dos motores foram impressos, enquanto que as peças necessárias para suportar as lagartas (Figura 4.16d) foram maquinadas numa placa de acrílico de 5 mm utilizando uma máquina de *Computer Numeric Control* (CNC).

O suporte do motor tem as dimensões necessárias para o correto posicionamento do motor na estrutura da lagarta, com a mesma furação dos motores para que os mesmos possam ser apertados e tem ainda um buraco na parte de trás onde é colocado o suporte do *encoder*.

O suporte do *encoder* tem a mesma forma da PCB do *encoder*, com alguma folga para permitir que a resina epóxi isole completamente a PCB. Tem ainda um furo na parte posterior para a passagem do cabo de alimentação e sinal do *encoder* e dois pinos para manter a PCB no sítio.

O suporte do íman serve apenas para manter o íman no eixo do motor.

Nas peças necessárias para o suporte das lagartas, foi utilizada uma placa de acrílico de 5 mm pela disponibilidade da mesma no laboratório, pelo peso e pela resistência que era suficiente para a função pretendida.

Para facilitar a implementação, cortou-se uma placa de poli-carbonato de 8 mm para ligar as duas lagartas e suportar todos os componentes elétricos do sistema, que no futuro podem ser colocados dentro de um compartimento estanque.

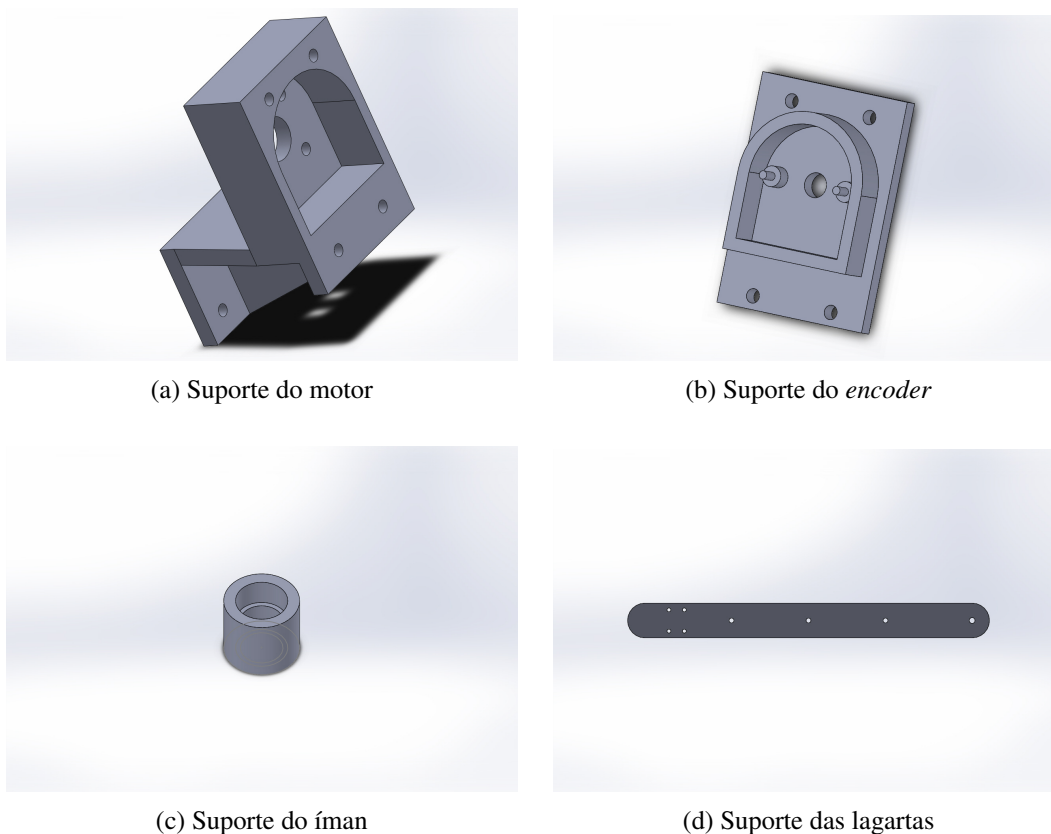


Figura 4.16: Peças desenhadas em SolidWorks.

4.3 Integração

Selecionados todos os componentes necessários para a implementação de um sistema (Figura 4.17) capaz de cumprir com os requisitos definidos, procedeu-se à preparação dos mesmos para poderem ser instalados, foi ainda necessário programar alguns dos componentes como foi o caso dos ESC e dos microprocessadores.

A nível de hardware, foi apenas necessário preparar e instalar os componentes na estrutura final. Com todos os componentes preparados para serem integrados em conjunto, procedeu-se à montagem do protótipo utilizando elementos resistentes à corrosão, como parafusos, porcas e anilhas de inox e ainda cantoneiras de alumínio. A montagem realizou-se numa placa de polycarbonato, sem que os componentes elétricos sensíveis à água estejam protegidos, esta decisão foi tomada devido à falta de tempo.

O protótipo final tem 41 cm de comprimento, 29,5 cm de largura, 11 cm de altura e com 2,8 kg de peso. As lagartas tem uma altura máxima de 9 cm, com 38 cm de comprimento em contacto com o solo e uma distância entre as lagartas é igual à 0,245 m.

Nas Figuras 4.18, 4.19 e 4.20 podemos observar o protótipo final.

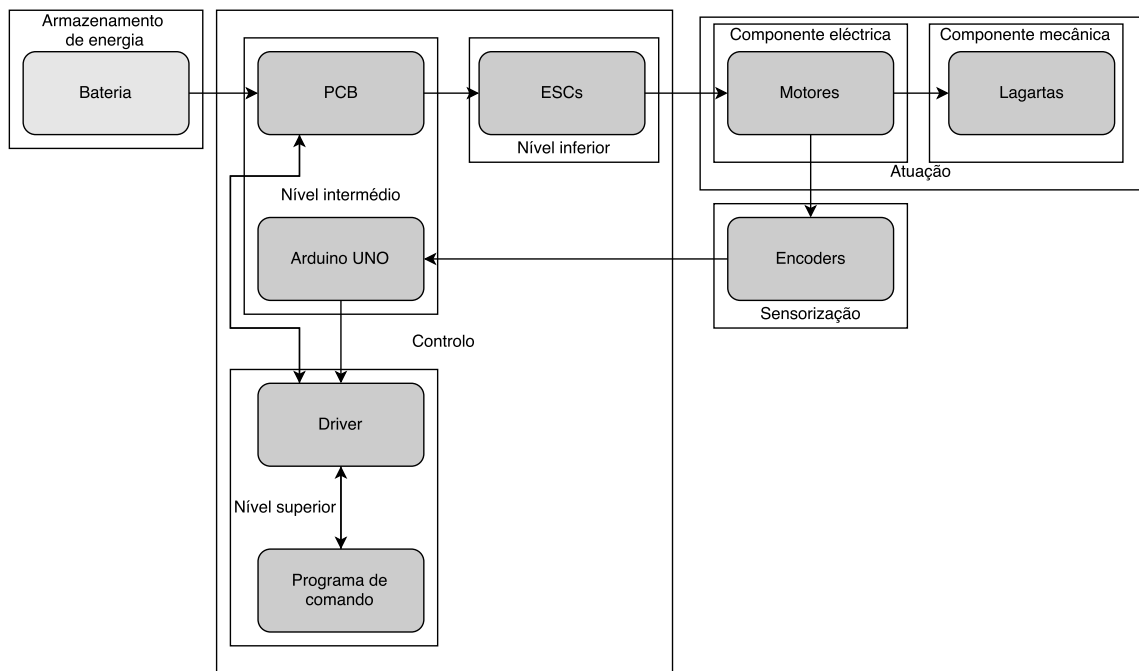


Figura 4.17: Diagrama do sistema final.

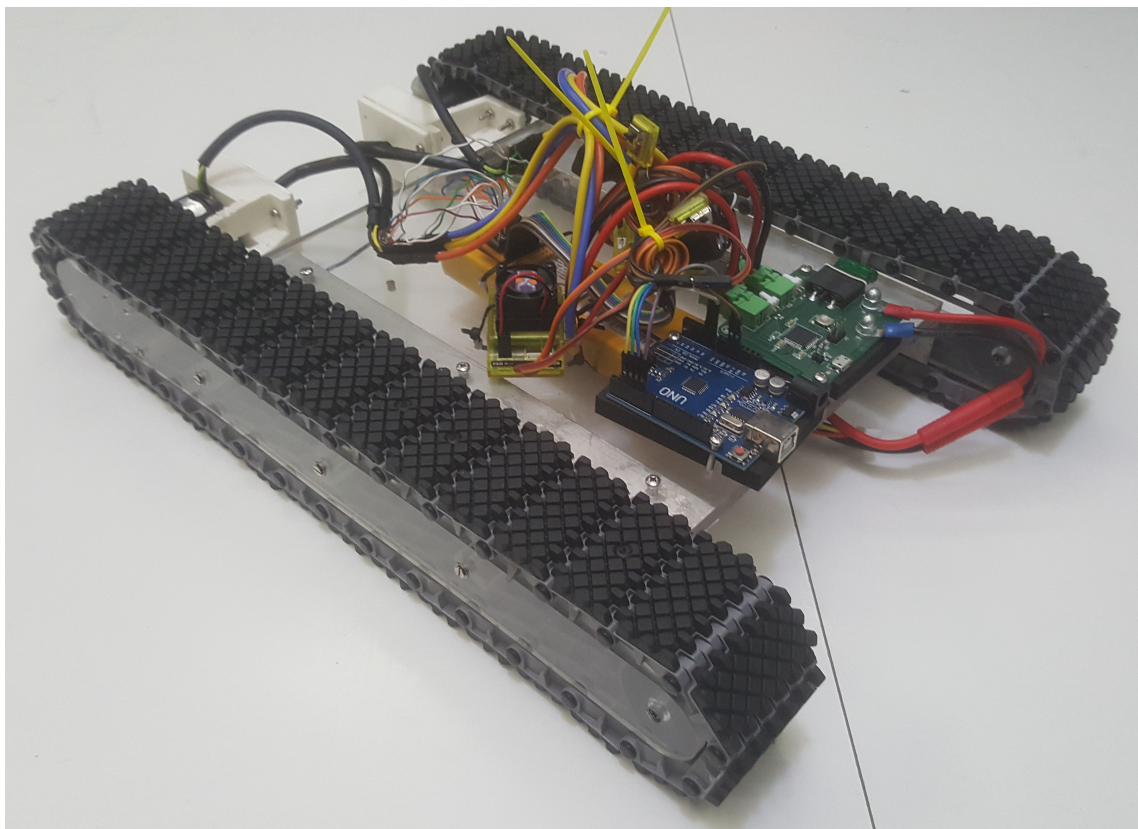


Figura 4.18: Vista em perspectiva do protótipo.

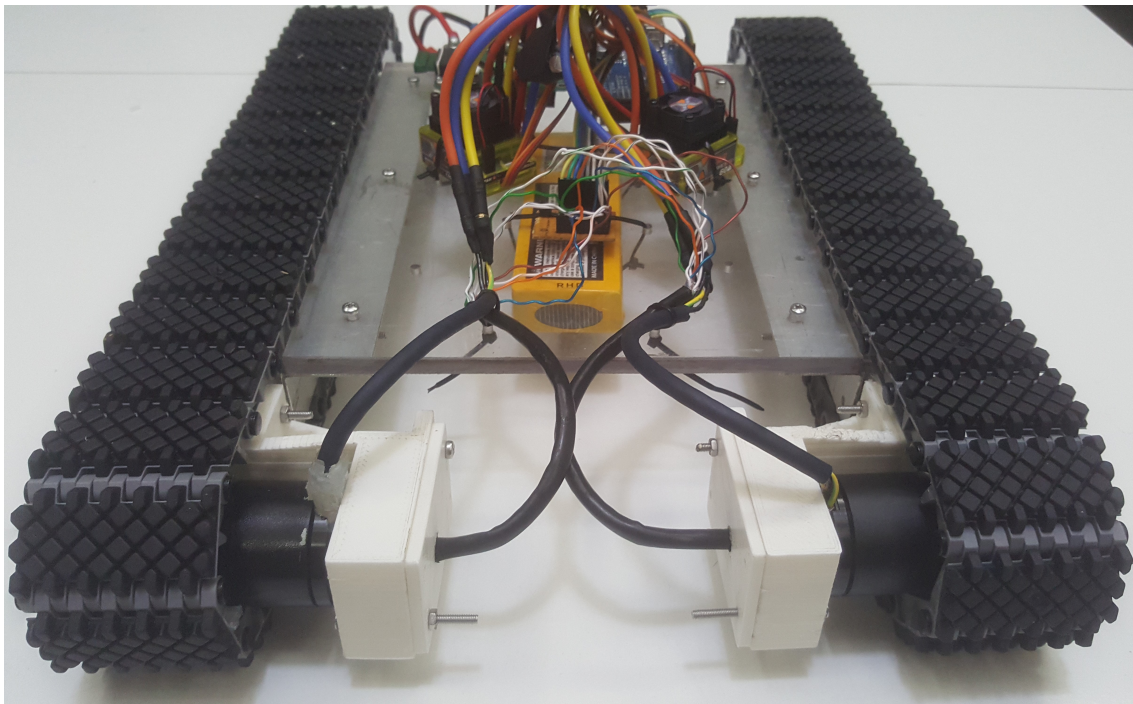


Figura 4.19: Vista frontal do protótipo.

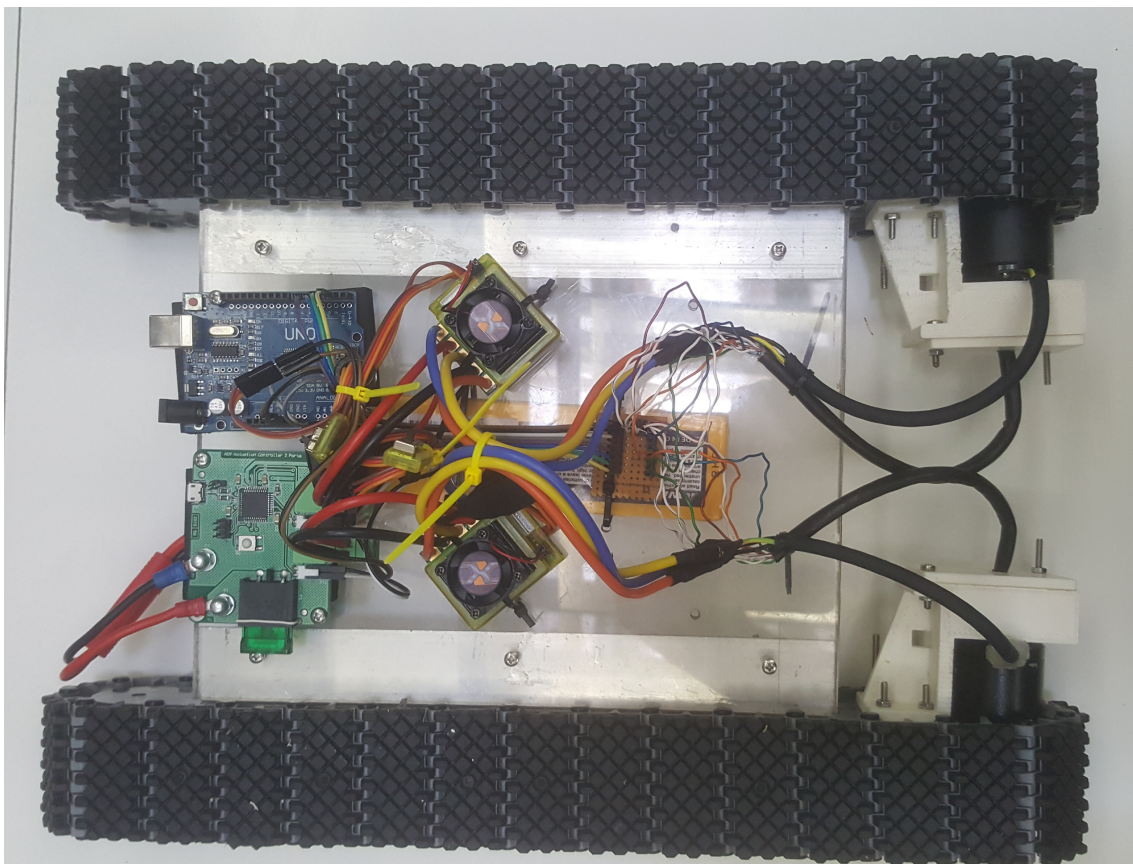


Figura 4.20: Vista superior do protótipo.

Capítulo 5

Software

A nível de software, como já foi referido, foi necessário programar diversos componentes como os ESC e os microprocessadores. Foi ainda necessário desenvolver uma aplicação para correr no nível superior de controlo.

Como os componentes que necessitam de ser programados e que irão correr software são elementos dos subsistemas de controlo, a sua implementação foi dividida pelos três níveis de controlo existentes.

5.1 Nível inferior

No nível inferior de controlo, é necessário programar e calibrar os ESC.

Os ESC seleccionados têm também a particularidade de poderem ser programados via USB com um adaptador (Figura 5.1). No manual do utilizador [30] existe uma explicação de como alterar os parâmetros e ainda o que resulta da sua alteração. Na Tabela 5.1 é possível observar os parâmetros disponíveis, com a configuração de fábrica e a configuração final.

Tabela 5.1: Parâmetros para programação dos ESC.

Parâmetro	Configuração inicial	Configuração final
<i>Running mode</i>	<i>Forward with Brake</i>	<i>Foward/Reverse</i>
<i>Drag brake force</i>	<i>10%</i>	<i>10%</i>
<i>Low voltage cut-off threshold</i>	<i>3.0V/cell</i>	<i>3.0V/cell</i>
<i>Start mode (punch)</i>	<i>Level VII</i>	<i>Level X</i>
<i>Max brake force</i>	<i>75%</i>	<i>75%</i>
<i>Max reverse force</i>	<i>25%</i>	<i>100%</i>
<i>Initial brake force</i>	<i>0%</i>	<i>0%</i>
<i>Neutral range</i>	<i>6%</i>	<i>6%</i>
<i>Timing</i>	<i>15°</i>	<i>15°</i>
<i>Over-heat protection</i>	<i>Enable</i>	<i>Enable</i>

Concluída a programação dos ESC foi necessário proceder à calibração. Foi seguido o procedimento indicado no manual do utilizador [30]. Foi necessário indicar aos ESC os valores em



Figura 5.1: O adaptador para ligar o ESC ao computador.

micro-segundos para os diferentes pontos fundamentais para o funcionamento dos ESC, o ponto neutro (1500 ms), o ponto máximo positivo (2000 ms) e o ponto máximo negativo (1000 ms).

Utilizando um Arduino UNO conectado ao fio de sinal e ao fio de massa dos ESC e a biblioteca *Servo.h* foi possível enviar para os ESC os valores pretendidos. Estes valores eram enviados para o Arduino através da comunicação por porta série no momento indicado pelo manual do utilizador para a correta programação dos ESC.

5.2 Nível intermédio

O nível intermédio de controlo é responsável por gerar os sinais de controlo do nível inferior na forma de sinais PWM e ainda por processar os sinais do *encoder*. Este nível está dividido em dois microprocessadores, o microprocessador na PCB do laboratório e o microprocessador do Arduino UNO. Cada um dos microprocessadores irá desempenhar diferentes funções, um o controlo dos ESC e o outro o processamento dos *encoder*, o que implica que cada um tenha um código diferente consoante as funções desempenhadas.

5.2.1 PCB do laboratório

Como a principal função desta PCB é a geração dos sinais para o controlo dos ESC, e ainda pelo facto da PCB ter sido desenhada de raiz pelo laboratório, levou a que fosse implementado o código desenvolvido e disponibilizado pelo laboratório para realizar essa mesma função.

Este código está preparado para receber mensagens através da comunicação efetuada por porta série. Estas mensagens são enviadas pelo nível superior e estão estruturadas segundo um protocolo desenvolvido pelo laboratório.

A PCB tem também um sensor de corrente por cada ESC, pelo que o código tem também a função de ler a informação da corrente que cada ESC está a utilizar e enviar essa informação através da comunicação por porta série numa mensagem segundo o protocolo anterior.

Na Figura 5.2 é possível observar o diagrama de estados do código que corre na PCB.

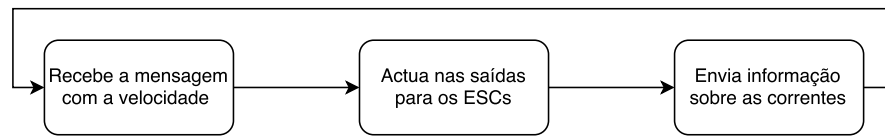


Figura 5.2: O diagrama de estados da PCB.

5.2.2 Arduino

O software implementado no Arduino tem várias funções como o processamento dos sinais dos *encoder*, a geração dos sinais UVW para os ESC com base num sinal absoluto da posição dos *encoder* (*Vout*) e a comunicação por porta série dos valores da distância percorrida por cada lagarta.

O código implementado funciona com base em *interrupts*, que correm em paralelo com a função principal. A utilização de *interrupts* é necessária devido à imprevisibilidade dos impulsos dos sinais A e B dos *encoder*, o que significa que sempre que for detetada uma variação no valor digital dos sinais A ou B, o Arduino irá correr uma função para contar o número de impulsos ocorridos.

Na função principal do programa o Arduino irá correr as funções para gerar os sinais UVW (Figura 5.3), calcular as distâncias percorridas por cada lagarta com base no número de impulsos e enviar essas distâncias por porta série.

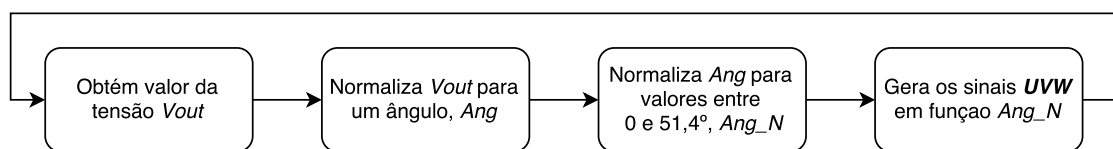


Figura 5.3: Sequência da função UVW do Arduino.

Com base nos sinais *Vout* dos *encoder* o Arduino irá gerar os sinais UVW. O *encoder* tem a possibilidade de definir a escala de *Vout*, isto é, definir o valor máximo e valor mínimo. Para obter a maior resolução possível com o Arduino, o valor máximo é 5 V e o valor mínimo é 0 V. Os sinais UVW dependem da posição do motor, como o motor tem 14 polos, cada sinal terá um período correspondente a uma rotação de $51,4^\circ$. Numa rotação completa do motor o sinal terá sete períodos. Para maior facilidade da implementação do código, o valor do ângulo é normalizado para o intervalo entre 0 e $51,4^\circ$ e com base nesse valor são gerados os sinais UVW.

Para o cálculo das distâncias é necessário primeiro calcular a constante para a relação entre o número de impulsos do *encoder* e a distância percorrida pelo robô. Usando as equações da odometria é possível calcular a constante. Com base na constante, é possível calcular a distância percorrida por cada lagarta. Com as distâncias percorridas por cada lagarta calculadas, estas são enviadas através da comunicação por porta série, segundo o mesmo protocolo desenvolvido pelo laboratório já referido anteriormente, em que a mensagem é a distância.

5.3 Nível superior

A este nível foram implementadas as funções da odometria e as manobras. É necessário ainda que tenha comunicação com o nível intermédio de controlo. Por uma questão de facilidade e disponibilidade, este nível funciona num PC baseado em Linux.

Como no nível intermédio foram utilizados protocolos desenvolvidos pelo laboratório para facilitar a comunicação através de mensagens, foi necessário usar esses mesmos protocolos neste nível para garantir a compatibilidade. Utilizando os exemplos de software fornecidos pelo laboratório foi possível criar as mensagens de comunicação com a PCB com a velocidade pretendida nos motores. O software disponibilizado está dividido em duas parte, o *driver* e o comando (Figura 5.4).

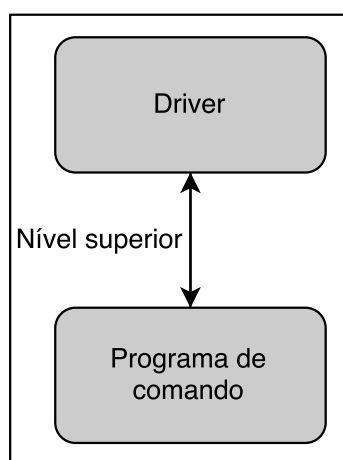


Figura 5.4: O nível superior de controlo.

5.3.1 *Driver*

O *driver* é responsável pela ligação com a PCB. É utilizado um ficheiro de configuração com todos os parâmetros necessários para o funcionamento do *driver*, como por exemplo, a porta série que é usada pela placa, a porta série que é usada pelo Arduino, as portas de comunicação com o programa de comandos e ainda alguns parâmetros necessários para a criação das ordens para a placa. Terminada a configuração do *driver*, o mesmo passa a criar as mensagens com a velocidade pretendida com base na informação do comando. Ao mesmo tempo, através da criação de três *threads* é possível enviar as mensagens criadas para a PCB e receber as mensagens da PCB e do Arduino.

5.3.2 Programa de controlo

Com base nas funções utilizadas no programa de envio de comandos para o *driver*, foram desenvolvidas funções para diferentes tipos de controlo do robô, se manual ou automático. No controlo manual é possível escolher diretamente a velocidade pretendida em cada um dos motores.

No modo automático já não é possível escolher diretamente a velocidade pretendida, apenas é possível indicar um ponto de destino para o robô, com as coordenadas x , y e θ . Para que a função seja implementada é necessário saber a posição atual do robô. É portanto necessário implementar as equações da odometria para calcular a posição do robô com base na distância percorrida por cada lagarta, recebida na mensagem do Arduino reencaminhada pelo *driver*. Para garantir que todas as mensagens reencaminhadas pelo *driver* são recebidas corretamente e não se perdem é utilizada uma *thread*, onde também são implementadas as equações da odometria.

A comunicação entre o *driver* e o programa de controlo é efetuada através de *sockets*. É necessário portanto configurar ambas as partes com os números dos portos para comunicação nos dois sentidos, pois não é possível dois processos diferentes utilizarem o mesmo porto.

5.3.3 Teste com os microprocessadores

Utilizando o código anteriormente desenvolvido, foi testada a capacidade de enviar ordens do PC para a PCB e posteriormente verificar se a PCB gerava os sinais para os ESC. Utilizando um Arduino verificou-se se os mesmos estavam de acordo com o pretendido, o que significava que a PCB estava a funcionar corretamente e que os comandos enviados na mensagem desde o programa de comando estavam a ser passados para o *driver* e posteriormente enviados para a PCB.

Foi também testada a capacidade de detetar os impulsos dos *encoder* usando os *interrupts* criados. Após verificar que os mesmos estavam a ser detetados, foi verificado se essa informação chegava ao programa de controlo, na mensagem criada pelo Arduino, que era enviada para o *driver* e reencaminhada para o programa de controlo.

Em ambos os testes foi também testada a comunicação por porta série entre os diferentes níveis de controlo e ainda se o protocolo de criação das mensagens estava a ser utilizado corretamente.

5.3.4 Odometria

Nesta solução, para efeitos de cálculos da odometria, considera-se um caso de tração diferencial, em que o robô tem duas lagartas. É fundamental para implementar a odometria a implementação de várias equações (Figura 5.5).

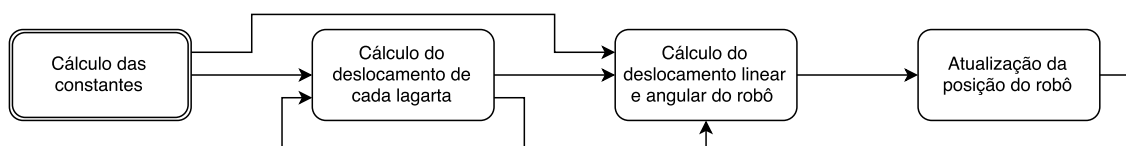


Figura 5.5: O processo para obter a posição através da odometria.

Inicialmente é necessário calcular experimentalmente a relação entre os impulsos do *encoder* e o deslocamento.

Com a relação calculada, pode-se então calcular a distância percorrida por cada lagarta com a equação 5.1.

$$\begin{cases} \Delta Dist_R = Ki * N_R \\ \Delta Dist_L = Ki * N_L \end{cases} \quad (5.1)$$

em que,

- Ki é a relação entre os impulsos e o deslocamento.
- $\Delta Dist_R$ é a distância percorrida pela lagarta direita.
- $\Delta Dist_L$ é a distância percorrida pela lagarta esquerda.
- N é o número de impulsos detetados.

Com as distâncias percorridas por ambas as rodas, é calculado o deslocamento do ponto central do robô utilizando as equações 5.2.

$$\begin{cases} \Delta D = \frac{\Delta Dist_R + \Delta Dist_L}{2} \\ \Delta \theta = \frac{\Delta Dist_R - \Delta Dist_L}{b} \end{cases} \quad (5.2)$$

em que,

- ΔD é a distância percorrida pelo robô.
- $\Delta \theta$ é a variação angular realizada pelo robô.
- b é a distância entre as rodas.

Por último, é atualizada a posição do robô com as equações 5.3.

$$\begin{cases} x_i = x_{i-1} + \Delta D * \cos(\theta_i + \frac{\Delta \theta_i}{2}) \\ y_i = y_{i-1} + \Delta D * \sin(\theta_i + \frac{\Delta \theta_i}{2}) \\ \theta_i = \theta_{i-1} + \Delta \theta_i \end{cases} \quad (5.3)$$

em que,

- θ_i é o ângulo do robô com o eixo x do referencial com origem na posição inicial do robô.
- x_i é a coordenada x do robô.
- y_i é a coordenada y do robô.

5.3.5 Função GotoXY

Com o sistema capaz de implementar a odometria e conseqüentemente ter a percepção da sua localização é possível implementar um conjunto de manobras automáticas, como é o caso da função GotoXY [9], na Figura 5.6 é possível observar o diagrama de estados da função em que as variáveis,

- *linear_vel* corresponde à velocidade linear.
- *ang_vel* corresponde à velocidade angular.
- *erro_dist* corresponde ao erro de distância ao ponto pretendido

- *erro_theta* corresponde ao erro absoluto entre o ângulo do robô e o destino.
- *erro_theta_f* corresponde ao erro absoluto entre o ângulo do robô e o ângulo final.
- *DIST_DA* corresponde à distância à qual o robô começa a desacelerar a velocidade linear.
- *TOL_FINDIST* corresponde à tolerância pretendida na distância ao destino.
- *THETA_DA* corresponde à distância à qual o robô começa a desacelerar a velocidade angular.
- *TOL_FINTHETA* corresponde à tolerância pretendida no ângulo do robô.
- *MAX ETF* corresponde ao erro na direção em que o robô começa a deslocar-se.
- *HIST ETF* corresponde ao erro na direção em que o robô para e roda novamente para o destino.
- *DIST_NEWPOSE* e *THETA_NEWPOSE* correspondem a uma nova posição final.

Esta função tem como estado inicial a posição do robô. Dependendo do ponto de destino, o robô roda, com a aplicação de velocidade angular, na direção do destino até que a direção do robô esteja dentro do intervalo pretendido. Com o robô alinhado na direção do destino, o robô começa a deslocar-se com uma velocidade linear constante e com uma velocidade angular proporcional ao erro da direção do robô para o destino, isto é, quanto maior for esse erro maior será a velocidade angular para corrigir a direção. Quando o robô se aproxima do destino, a velocidade linear é reduzida para garantir uma maior precisão, até parar nas coordenadas x e y do destino. Apenas falta rodar para o ângulo θ pretendido, é necessário então aplicar velocidade angular e quando este se aproxima do ângulo θ pretendido, reduz essa velocidade para obter uma maior precisão. Quando o ângulo θ é atingido o robô para e a função termina, no entanto é possível a qualquer momento alterar o ponto do destino, fazendo com que o robô volte ao início do procedimento.

Com esta função é possível no futuro criar uma sequência de manobras, basta para isso indicar os diversos pontos que se pretende que o robô passe e assim criar uma manobra mais complexa.

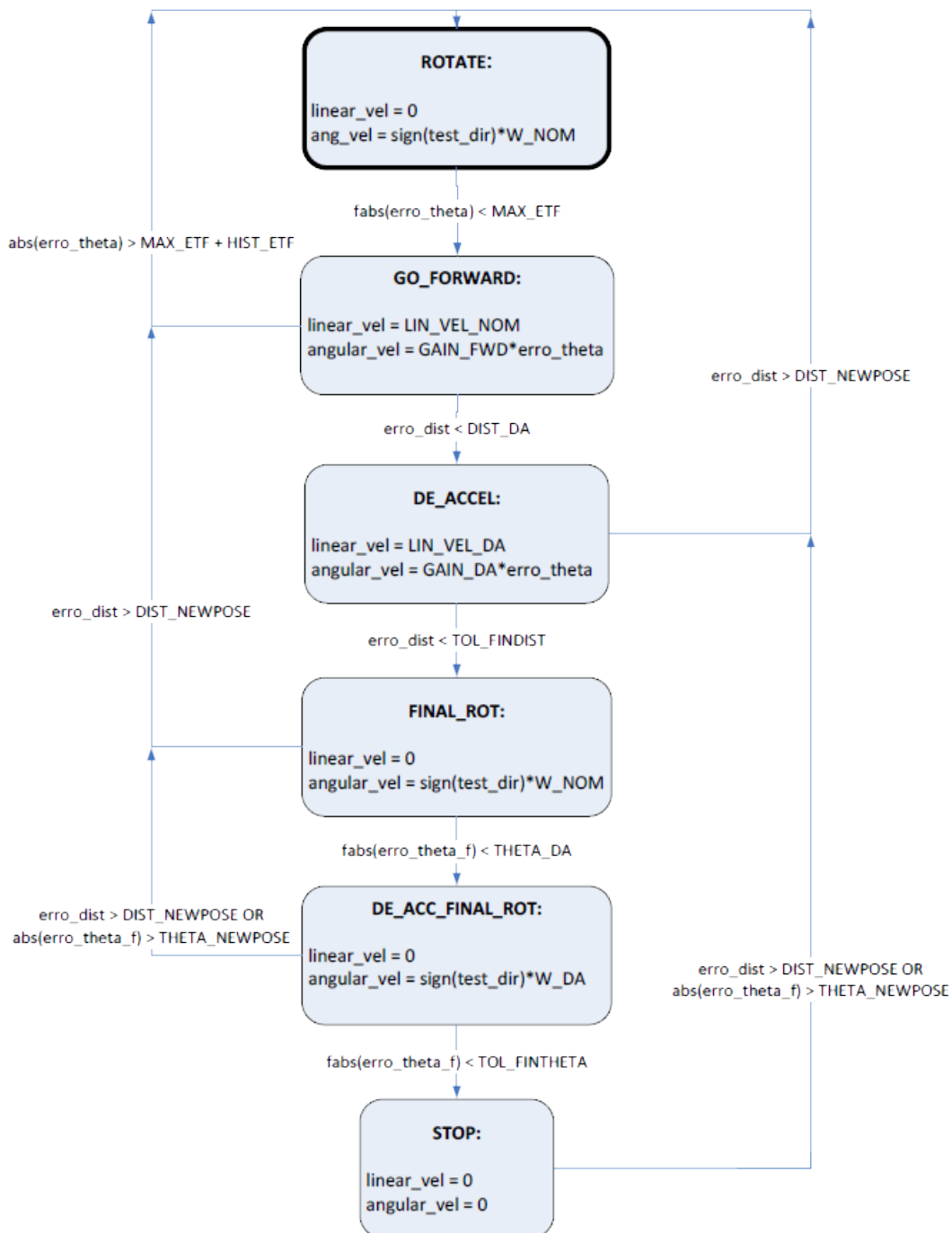


Figura 5.6: Diagrama de estados da função gotoXY. Fonte:[9].

Capítulo 6

Testes e Resultados

Alguns dos testes realizados tinham como objetivo a caracterização dos componentes, pelo que os resultados foram integrados no Capítulo 4. Neste Capítulo são referidos os testes e resultados obtidos após terminada a escolha e a caracterização dos diversos componentes e após os mesmos terem sido implementados, procedeu-se aos testes do sistema completo. Estes testes tinham como objetivo verificar se o sistema conseguia cumprir alguns dos requisitos, em particular se o robô se conseguia deslocar e ainda se o robô se conseguia localizar com base na odometria.

6.1 Locomoção

Com o sistema implementado e a funcionar procedeu-se à verificação se os motores tinham força suficiente para deslocarem o robô e ainda o deslocamento obtido utilizando os ESC em modo *sensorless*. No gráfico da Figura 6.1 é possível observar o deslocamento obtido em função do comando enviado.

Para este teste, o comando era enviado e após terminado o período definido de 3 segundos, o robô para. Como é possível verificar existe uma zona morta entre os comandos -20% e 20%, em que o robô não se movimenta. O robô apenas se movimenta com comandos superiores a 20% em valor absoluto. A partir dos comandos superiores a 25%, em valor absoluto, o deslocamento começa a tender para 1 m devido à configuração efetuada no *driver*.

6.2 Cálculo das constantes para a odometria

Para poder implementar a odometria é necessário primeiro determinar as constantes. A relação entre os impulsos do *encoder* e o deslocamento linear da lagarta e ainda a distância entre as lagartas.

Com o robô capaz de se movimentar, executou-se um conjunto de ordens básicas através da função de controlo manual apenas com velocidade linear para que o robô se deslocasse fisicamente 1 m em ambos os sentidos. Após o robô ter-se deslocado 1 m, o número de impulsos resultante era apontado. Na Tabela 6.1 é possível observar os valores obtidos.

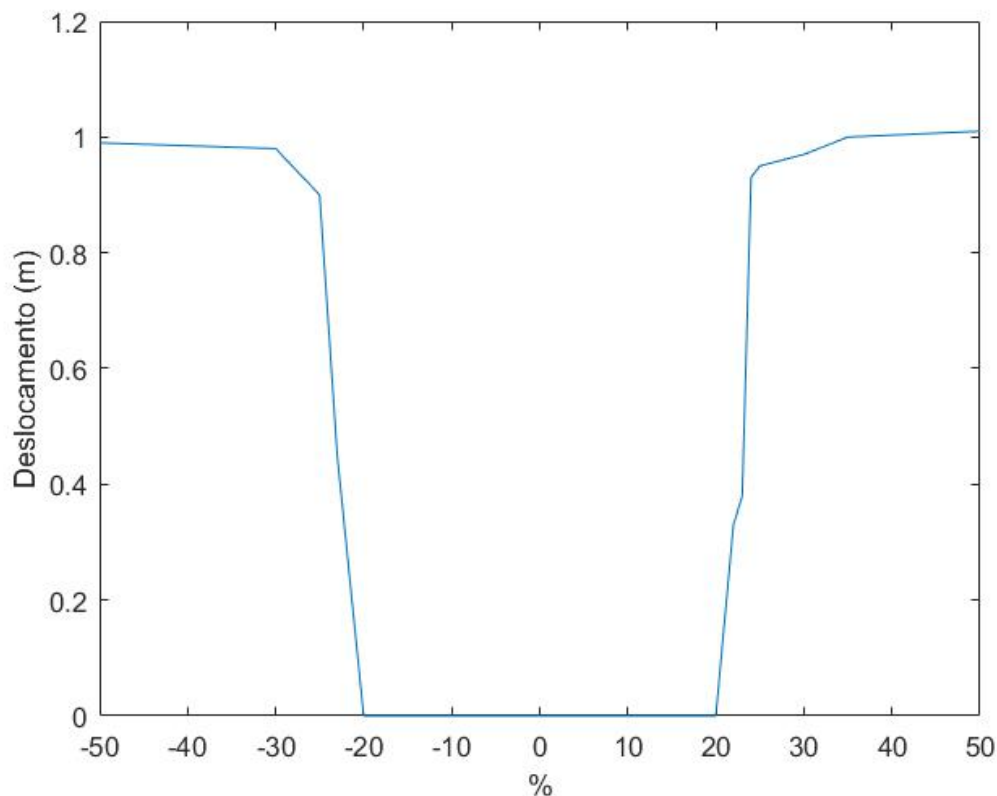


Figura 6.1: O deslocamento obtido em função do comando.

Tabela 6.1: Número de impulsos dos *encoder* com o deslocamento de 1 m.

Sentido do deslocamento	Número de impulsos	
	Esquerda	Direita
Positivo	11691	11612
Negativo	-12257	-12134
Positivo	12346	12306
Negativo	-12307	-12264
Positivo	12184	11871
Negativo	-12488	-12183

A diferença entre os valores ocorre devido à existência de vários erros detetados ao longo do teste. Erros como o deslizar das lagartas no início, o atrito nas lagartas não ser constante e ainda devido ao facto de ser necessário parar manualmente o robô no fim do percurso, o que não permite que a deslocação efetuada seja exatamente 1 m.

Para determinar a constante K_i , calculou-se o valor médio das leituras de ambos os sentidos, para cada lagarta. Com o valor médio de impulsos de cada lagarta em ambos os sentidos, foi calculado o valor médio entre a lagarta esquerda e direita para cada sentido e posteriormente foi calculado o valor médio absoluto entre ambos os sentidos. No final obteve-se o valor médio de 12137 impulsos por metro. No entanto para a odometria, a constante tem que ser metros por

impulso. Sendo necessário então calcular o inverso do resultado obtido. No final a constante Ki resultou em $8,2393 \times 10^{-5}$ m/ impulso.

A distância entre as lagartas é igual a 0,245 m.

Ambos os valores foram inseridos no código para a correta implementação da odometria.

6.3 Localização

Terminados os testes da odometria, foi realizado um conjunto de testes para testar a função GotoXY e ainda a localização relativa do robô. Estes testes consistiram no envio de ordens para o robô se deslocar para o ponto $(x, y, \theta) = (1, 0, 0)$. No gráfico da Figura 6.2 podemos observar o resultado obtido na realização de um teste.

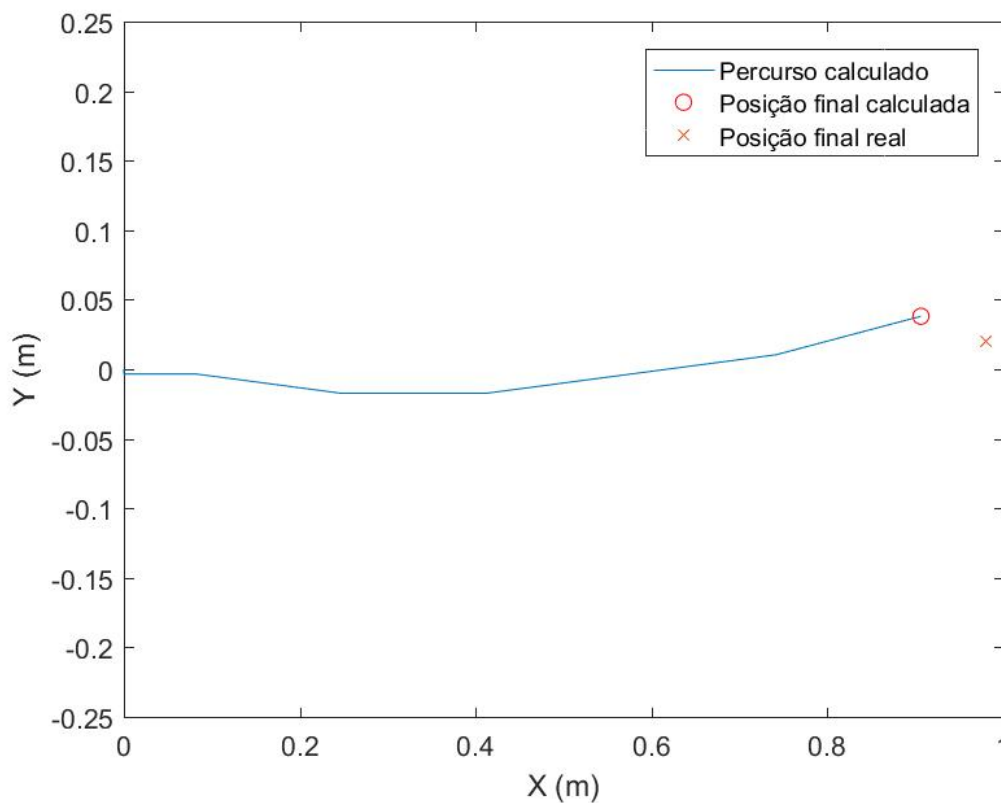


Figura 6.2: O percurso estimado e a posição real e estimada do robô.

Através da observação do gráfico é possível retirar várias conclusões. Em primeiro lugar, o robô não parou no ponto $(1, 0)$ devido às tolerâncias definidas na função GotoXY, que são elevadas para inicialmente garantir que o robô termina o movimento. A posição real não coincide com a posição calculada pela odometria, como seria de esperar, devido os erros na leitura dos valores dos *encoder* e ainda por fatores como a derrapagens das lagartas no arranque. É ainda possível observar que o robô não realizou uma trajetória linear como seria de esperar entre o ponto inicial e final devido a diferença, observada na realização dos testes, no atrito que cada lagarta oferecia, isto

é, uma lagarta movimenta-se com maior facilidade, e imprimindo o mesmo comando a ambas, esta irá rodar ligeiramente mais rápido, o que depois é corrigido pela função e que resulta na trajetória que é possível observar.

Todos estes erros podem ser melhorados através da realização de mais testes e do ajuste dos parâmetros da função GotoXY.

Capítulo 7

Conclusão

O sistema de locomoção desenvolvido ao longo do projeto tinha como propósito o deslocamento em superfícies subaquáticas como, por exemplo, o solo ou estruturas submarinas e ser capaz de calcular a sua posição relativa.

7.1 Cumprimento dos objetivos e requisitos

No final do projeto, o rover desenvolvido é capaz de se deslocar em superfícies secas e é capaz de calcular a sua posição relativa recorrendo a odometria cumprindo assim os principais objetivos definidos para o sistema. Outros objetivos foram também cumpridos como o projeto de manobras elementares através da função GotoXY.

Os requisitos definidos no Capítulo 3 são cumpridos com exceção da compatibilidade com o AUV SHAD e ainda o requisito de deslocar-se no solo subaquático. Estes requisitos não foram alcançados devido ao curto espaço de tempo disponível para desenvolver um design capaz de ser integrado com o SHAD e ainda porque o protótipo não tinha os componentes elétricos, como a bateria ou os microprocessadores, isolados do meio ambiente.

Em relação à capacidade do robô se localizar, esta capacidade foi desenvolvida e integrada mas no entanto não foi possível testar completamente todos os parâmetros para garantir que funciona corretamente.

7.2 Trabalho futuro

No futuro, é necessário criar um design que permita ao protótipo desenvolvido seja integrado com o SHAD ao mesmo tempo que isola todos os componentes necessários do contacto com a água. É necessário também realizar mais testes sobre a capacidade de localização e a função GotoXY, para otimizar e reduzir os erros detetados na realização dos testes. Uma possível melhoria é a integração de uma Raspberry PI para substituir o PC no nível superior de controlo e assim obter um sistema mais portátil e autónomo.

Para melhorar a capacidade de deslocação do robô, é necessário também mudar os ESC para o sistema *sensored*. Isto implica o sincronismo entre a posição do motor e a comutação das fases. Recorrendo a um osciloscópio e com o ajuste do incremento do valor do ângulo no código de geração dos sinais é possível observar para que incremento é obtido o sincronismo. Isto irá permitir um melhor funcionamento dos motores para as baixas rotações.

Outra possível melhoria é também a integração de uma caixa de transmissão entre cada motor e a lagarta para aumentar o binário fornecido e ao mesmo tempo diminuir a velocidade das lagartas em relação aos motores, pois estes motores funcionam melhor a altas velocidades.

Referências

- [1] Liam Paull, Sajad Saeedi, Mae Seto, e Howard Li. AUV Navigation and Localization: A Review. *IEEE Journal of Oceanic Engineering*, 39(1):131–149, Janeiro 2014. URL: <http://ieeexplore.ieee.org/document/6678293/>, doi:10.1109/JOE.2013.2278891.
- [2] C. Waldmann e M. Bergenthal. CMOVE-a versatile underwater vehicle for seafloor studies. Em *OCEANS 2010*, páginas 1–3. IEEE, 2010. URL: <http://ieeexplore.ieee.org/abstract/document/5664261/>.
- [3] F. Wenzhoefer, T. Wulff, S. Floegel, S. Sommer, e C. Waldmann. ROBEX-innovative robotic technologies for ocean observations, a deep-sea demonstration mission. Em *OCEANS 2016 MTS/IEEE Monterey*, páginas 1–8. IEEE, 2016. URL: <http://ieeexplore.ieee.org/abstract/document/7761215/>.
- [4] Bong-Huan Jun, Hyungwon Shim, Banghyun Kim, Jin-Yeong Park, Hyuk Baek, Seongyeol Yoo, e Pan-Mook Lee. Development of seabed walking robot CR200. Em *OCEANS-Bergen, 2013 MTS/IEEE*, páginas 1–5. IEEE, 2013. URL: <http://ieeexplore.ieee.org/abstract/document/6608164/>.
- [5] François Michaud, Dominic Létourneau, Martin Arsenault, Yann Bergeron, Richard Cadrin, Frédéric Gagnon, Marc-Antoine Legault, Mathieu Millette, Jean-François Paré, Marie-Christine Tremblay, Pierre Lepage, Yan Morin, Jonathan Bisson, e Serge Caron. Multi-Modal Locomotion Robotic Platform Using Leg-Track-Wheel Articulations. *Autonomous Robots*, 18(2):137–156, Março 2005. URL: <http://link.springer.com/10.1007/s10514-005-0722-1>, doi:10.1007/s10514-005-0722-1.
- [6] Blue Robotics. M200 motor documentation. Disponível em: <http://docs.bluerobotics.com/thrusters/motors/m200/>.
- [7] RLS Merilna tehnika d.o.o. Rmk4 evaluation board. Disponível em: <https://www.rls.si/rmk4-evaluation-board>.
- [8] RLS Merilna tehnika d.o.o. *AM4096 – 12 bit angular magnetic encoder IC Data sheet*, 2016. Disponível em: <https://www.rls.si/rmk4-evaluation-board>.
- [9] António Paulo Gomes Mendes Moreira. Slides das aulas teóricas da unidade curricular sistema robóticos autónomos. Disponível em: https://sigarra.up.pt/feup/pt/conteudos_geral.zip?pct_cont_grp_id=59231.
- [10] Freyr Hardarson. *Locomotion for difficult terrain*. Citeseer, 1998. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.31.5881&rep=rep1&type=pdf>.

- [11] Bruno Ferreira INESC-TEC. Proposta de dissertação / projecto, 2016. Disponível em: https://sigarra.up.pt/feup/pt/estagios_empresas.ver_dados_proposta?p_id=206646&p_processo=21673&p_inst_codigo=&pv_perfil=ALU&p_aluno_id=117590.
- [12] Peter Putz. Space robotics. *Reports on Progress in Physics*, 65(3):421, 2002. URL: <http://iopscience.iop.org/article/10.1088/0034-4885/65/3/202/meta>.
- [13] Junku Yuh. Design and control of autonomous underwater robots: A survey. *Autonomous Robots*, 8(1):7–24, 2000. URL: <http://link.springer.com/article/10.1023/A:1008984701078>.
- [14] Johann Borenstein, H. R. Everett, Liqiang Feng, e others. Where am I? Sensors and methods for mobile robot positioning. *University of Michigan*, 119(120):27, 1996. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.444.5014&rep=rep1&type=pdf>.
- [15] Paul A. Miller, Jay A. Farrell, Yuanyuan Zhao, e Vladimir Djapic. Autonomous Underwater Vehicle Navigation. *IEEE Journal of Oceanic Engineering*, 35(3):663–678, Julho 2010. URL: <http://ieeexplore.ieee.org/document/5546885/>, doi:10.1109/JOE.2010.2052691.
- [16] Marco Morgado, Paulo Oliveira, e Carlos Silvestre. Experimental evaluation of a USBL underwater positioning system. Em *ELMAR, 2010 proceedings*, páginas 485–488. IEEE, 2010. URL: <http://ieeexplore.ieee.org/abstract/document/5606116/>.
- [17] S. M. Smith e D. Kronen. Experimental results of an inexpensive short baseline acoustic positioning system for AUV navigation. Em *OCEANS'97. MTS/IEEE Conference Proceedings*, volume 1, páginas 714–720. IEEE, 1997. URL: <http://ieeexplore.ieee.org/abstract/document/634454/>.
- [18] Tim Bailey e Hugh Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006. URL: <http://ieeexplore.ieee.org/abstract/document/1678144/>.
- [19] Jeffrey Aguilar, Tingnan Zhang, Feifei Qian, Mark Kingsbury, Benjamin McInroe, Nicole Mazouchova, Chen Li, Ryan Maladen, Chaohui Gong, Matt Travers, Ross L Hatton, Howie Choset, Paul B Umbanhowar, e Daniel I Goldman. A review on locomotion robophysics: the study of movement at the intersection of robotics, soft matter and dynamical systems. *Reports on Progress in Physics*, 79(11):110001, Novembro 2016. URL: <http://stacks.iop.org/0034-4885/79/i=11/a=110001?key=crossref.ca2966c57b06c47b4fe1725316799370>, doi:10.1088/0034-4885/79/11/110001.
- [20] Frank Wenzhöfer, Johannes Lemburg, Michael Hofbauer, Sascha Lehmenhecker, e Paul Färber. TRAMPER. Em *OCEANS 2016 MTS/IEEE Monterey*, páginas 1–6. IEEE, 2016. URL: <http://ieeexplore.ieee.org/abstract/document/7761217/>.
- [21] Bong-Huan Jun, Hyungwon Shim, Jin-Yeong Park, Banghyun Kim, e Pan-Mook Lee. A new concept and technologies of multi-legged underwater robot for high tidal current environment. Em *Underwater Technology (UT), 2011 IEEE Symposium on and 2011 Workshop on Scientific Use of Submarine Cables and Related Technologies (SSC)*, páginas 1–5. IEEE, 2011. URL: <http://ieeexplore.ieee.org/abstract/document/5774159/>.

- [22] B. H. Jun, H. Shim, B. Kim, J. Y. Park, H. Baek, P. M. Lee, W. J. Kim, e Y. S. Park. Preliminary design of the multi-legged underwater walking robot CR200. Em *OCEANS, 2012-Yeosu*, páginas 1–4. IEEE, 2012. URL: <http://ieeexplore.ieee.org/abstract/document/6263600/>.
- [23] Uluc Saranlı, Martin Buehler, e Daniel E. Koditschek. RHex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001. URL: <http://journals.sagepub.com/doi/abs/10.1177/02783640122067570>.
- [24] Shiwu Zhang, Xu Liang, Lichao Xu, e Min Xu. Initial Development of a Novel Amphibious Robot with Transformable Fin-Leg Composite Propulsion Mechanisms. *Journal of Bionic Engineering*, 10(4):434–445, Outubro 2013. URL: <http://linkinghub.elsevier.com/retrieve/pii/S1672652913602474>, doi:10.1016/S1672-6529(13)60247-4.
- [25] Gregory Dudek, Philippe Giguere, Chris Prahacs, Shane Saunderson, Junaed Sattar, Luz-Abril Torres-Mendez, Michael Jenkin, Andrew German, Andrew Hogue, Arlene Ripsman, e others. Aqua: An amphibious autonomous robot. *Computer*, 40(1), 2007. URL: <http://ieeexplore.ieee.org/abstract/document/4069194/>.
- [26] Carlos Filipe Silva Gonçalves. Projeto e Desenvolvimento de um Veículo Submarino Autônomo. 2016. URL: <https://repositorio-aberto.up.pt/handle/10216/85196>.
- [27] OpenROV. Trident underwater drone. Disponível em: <https://www.openrov.com/products/trident/#specs>.
- [28] Bhim Singh e Sanjeev Singh. State of the art on permanent magnet brushless DC motor drives. *journal of power electronics*, 9(1):1–17, 2009. URL: <http://www.dbpia.co.kr/Journal/ArticleDetail/NODE01115529>.
- [29] A. Kiruthika, A. Albert Rajan, e P. Rajalakshmi. Mathematical modelling and speed control of a sensed brushless DC motor using intelligent controller. Em *Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), 2013 International Conference on*, páginas 211–216. IEEE, 2013. URL: <http://ieeexplore.ieee.org/abstract/document/6528495/>.
- [30] HobbyKing. User manual xerun 60a/120a. Disponível em: <https://hobbyking.com/media/file/1023053407X422598X57.pdf>.