

**ESCOLA NAVAL**

**DEPARTAMENTO DE FORMAÇÃO DA CLASSE DE  
ENGENHEIROS NAVAIS RAMO DE ARMAS E  
ELETRÓNICA**



**SISTEMA DE POSICIONAMENTO AUTÓNOMO**

**Hugo Daniel Cruz Simões**

MESTRADO EM CIÊNCIAS MILITARES NAVAIS  
CLASSE DE ENGENHEIROS NAVAIS RAMO DE ARMAS E  
ELETRÓNICA

2013

**ESCOLA NAVAL**

**DEPARTAMENTO DE FORMAÇÃO DA CLASSE DE  
ENGENHEIROS NAVAIS RAMO DE ARMAS E  
ELETRÓNICA**



**SISTEMA DE POSICIONAMENTO AUTÓNOMO**

**Hugo Daniel Cruz Simões**

**MESTRADO EM CIÊNCIAS MILITARES NAVAIS  
CLASSE DE ENGENHEIROS NAVAIS RAMO DE ARMAS E  
ELETRÓNICA**

Professor Orientador: Prof. Doutor Victor Lobo

2013

**ESCOLA NAVAL**

**DEPARTAMENTO DE FORMAÇÃO DA CLASSE DE  
ENGENHEIROS NAVAIS RAMO DE ARMAS E  
ELETRÓNICA**

**SISTEMA DE POSICIONAMENTO AUTÓNOMO**

MESTRADO EM CIÊNCIAS MILITARES NAVAIS  
CLASSE DE ENGENHEIROS NAVAIS RAMO DE ARMAS E  
ELETRÓNICA

O Mestrando,

*(assinado no original)*

---

ASPOF EN-AEL Cruz Simões

O Orientador

*(assinado no original)*

---

Prof. Doutor Victor Lobo

# AGRADECIMENTOS

---

A elaboração deste projeto teve a duração de um ano letivo e teve a intervenção de várias pessoas, cada um com o seu contributo. De entre as quais gostava de demonstrar o meu agradecimento:

Ao meu orientador Professor Doutor Victor Sousa Lobo por me ter acompanhado e ajudado a realizar esta dissertação.

Ao CFR EN-AEL Ribeiro Correia, não só pela preocupação ao longo destes anos de Escola Naval, mas também pelo acompanhamento, ajuda e disponibilidade na realização desta dissertação.

Ao ASPOF EN-AEL Jorge Jesus com o qual passei bastante tempo a trabalhar, e nos ajudámos mutuamente. E aos restantes camaradas do Curso Padre Fernando Oliveira por partilharem estes cinco anos de Escola Naval.

Ao Departamento de Formação de Engenharia Naval de Armas e Eletrónica da Escola Naval, por todo o apoio prestado.

À minha família e amigos que me apoiaram durante este período em que estive envolvido na realização desta dissertação de mestrado, e compreenderam os momentos em que não estive presente.

À minha namorada, Ana Bicho, por todo o apoio e dedicação neste período.

Ao meu irmão, João Simões, por toda a sua disponibilidade durante todo este projeto, em especial, durante a realização dos testes ao protótipo.

A todos aqueles que ao longo destes anos me apoiaram e tornaram esta dissertação possível. Aqui fica um sentido muito obrigado.

# RESUMO

---

O Projeto de Desenvolvimento apresentado nesta Dissertação surge na sequência de contatos realizados junto da Federação Portuguesa de Vela com o objetivo de construção de um protótipo de boias com posicionamento automático.

O objetivo deste protótipo é, de uma forma autónoma, colocar uma boia em alto mar, e que esta se mantenha numa posição pré-definida. Para responder a tal necessidade, procurou-se desenvolver um sistema de posicionamento autónomo com base num recetor GPS, ou seja, que tenha a capacidade de posicionar-se de forma independente sem qualquer tipo de controlo remoto. Os desenvolvimentos alcançados com este projeto vão servir de base a outros projetos de investigação atualmente a decorrer no âmbito do Centro de Investigação Naval (CINAV).

Esta dissertação de mestrado relata os procedimentos efetuados durante o desenvolvimento do sistema de posicionamento, construção da placa de controlo e escolha de algoritmo de posicionamento. São também descritos os testes efetuados, em piscina e rio. Por fim, são feitas algumas considerações e sugestões, realçando os aspetos positivos e os negativos.

**Palavras-Chave:** Marinha Portuguesa, Escola Naval, Boia, Posicionamento, Station Keeping.

# ABSTRACT

---

The Developed Project presented in this dissertation, arises from contacts made with the Portuguese Sailing Federation, with the aim of building a prototype buoy with automatic positioning.

The goal of this prototype is to place a buoy at the sea, and autonomously, remain at a pre-defined position. To answer this need, based on a GPS receiver, it became necessary to develop an autonomous positioning system, in other words, the buoy needed the ability to position itself without any support of a remote control. The developments achieved by this project will form the basis of other research projects currently underway within the Centro de Investigação Naval (CINAV).

This dissertation describes the procedures performed during the development of the positioning systems, the building of the control board and choice of the positioning algorithm. Also described are the tests performed in pool and river. To conclude, there are some considerations and suggestions, highlighting the positive and negative aspects.

**Keywords:** Portuguese Navy, Naval Academy, Buoy, Positioning, Station Keeping.

# ÍNDICE

---

AGRADECIMENTOS .....	iv
RESUMO .....	v
ABSTRACT .....	vi
ÍNDICE.....	vii
LISTA DE FIGURAS .....	x
LISTA DE TABELAS .....	xii
LISTA DE EQUAÇÕES .....	xiii
LISTA DE ABREVIATURAS.....	xiv
CAPÍTULO 1 - INTRODUÇÃO.....	15
1.1. ENQUADRAMENTO .....	15
1.2. JUSTIFICAÇÃO DO TEMA .....	16
1.3. OBJETIVOS.....	16
1.4. REQUISITOS.....	17
CAPÍTULO 2 - ESTADO DE ARTE .....	18
2.1. SISTEMAS DE POSICIONAMENTO DINÂMICO .....	19
2.1.1. POSICIONAMENTO GEOGRÁFICO ATRAVÉS DE REDE ACÚSTICA ..	21
2.1.2. POSICIONAMENTO GEOGRÁFICO BASEADO EM VISÃO E LASER ...	22
2.1.3. POSICIONAMENTO GEOGRÁFICO BASEADO EM GPS .....	23
2.1.4. ESCOLHA DO SISTEMA DE POSICIONAMENTO DINÂMICO .....	25
2.2. SISTEMA DE CONTROLO PID.....	26
2.2.1. EQUAÇÃO PID.....	26
2.2.2. CONTROLO EM MALHA FECHADA .....	27

2.2.3.	RESPOSTA PID .....	29
2.2.4.	LIMITADORES.....	32
2.2.5.	OUTROS ALGORITMOS.....	32
2.3.	<i>HARDWARE</i> .....	38
2.3.1.	<i>GLOBAL POSITIONING SYSTEM</i> .....	38
2.3.2.	SISTEMA DE COMANDO E CONTROLO.....	47
CAPÍTULO 3 - ARQUITETURA DO SISTEMA.....		52
3.1.	<i>HARDWARE</i> .....	52
3.2.	<i>SOFTWARE</i> .....	55
3.3.	<i>INTERFACE</i> .....	62
CAPÍTULO 4 – SIMULADOR E TESTES .....		64
4.1.	SIMULADOR .....	65
4.1.1.	IMPLEMENTAÇÃO DOS ALGORITMOS DE POSICIONAMENTO .....	68
4.1.2.	SIMULAÇÃO DO ALGORITMO <i>WAY-POINT</i> .....	70
4.1.3.	SIMULAÇÃO DO ALGORITMO <i>LINE-TRACKING</i> .....	72
4.1.4.	CONCLUSÕES.....	74
4.2.	TESTES.....	76
4.1.1	TESTES AO <i>RECETOR GPS LS20031</i> .....	77
4.1.2	TESTES AO CONTROLADOR PID .....	79
4.1.3	CONCLUSÃO .....	88
CAPÍTULO 5 - CONCLUSÃO.....		89
REFERÊNCIAS BIBLIOGRÁFICAS .....		92
APÊNDICE A – BIBLIOTECA DO ARDUINO .....		96
APÊNDICE B – ROTINAS DO ARDUINO .....		98
APÊNDICE C – CHECKLIST PARA TESTES.....		102

APÊNDICE D – CD-ROM ..... 103

# LISTA DE FIGURAS

---

FIGURA 1 – BOIA DE REGATA DE FORMA CILÍNDRICA. (MUNDO DA NÁUTICA, 2011). ....	15
FIGURA 2 – SISTEMA DE POSICIONAMENTO DINÂMICO UTILIZANDO SONO-BOIAS. (ALMEIDA, 2010).....	22
FIGURA 3 – <i>FANBEAM</i> , COMO SISTEMA DE POSICIONAMENTO DINÂMICO. (BUSCARIOLLO, 2008).....	23
FIGURA 4 – SISTEMA DE POSICIONAMENTO GEOGRÁFICO BASEADO EM GPS. ....	24
FIGURA 5 - CONTROLO EM MALHA FECHADA. (SILVA, 2000).....	27
FIGURA 6 – CONTROLADOR PROPORCIONAL. (GUERRA, 2009). ....	29
FIGURA 7 – CONTROLADOR PROPORCIONAL E DERIVATIVO. (GUERRA, 2009). ....	30
FIGURA 8 – CONTROLADOR PROPORCIONAL, INTEGRAL E DERIVATIVO. (GUERRA, 2009). ...	31
FIGURA 9 - RESPOSTA DO CONTROLADOR SEM SUSPENDER A AÇÃO INTEGRAL. (PALMA, 2008).....	32
FIGURA 10 - ORIENTAÇÃO DA BOIA, COM $V_X = V_Y = 0$ . (SANTOS, 2008, P.59). ....	33
FIGURA 11 - ORIENTAÇÃO DA BOIA, COM $V_X = 0$ ; $V_Y = 0.5$ . (SANTOS, 2008, P.59).....	34
FIGURA 12 - ORIENTAÇÃO DA BOIA, COM $V_X = 0.5$ ; $V_Y = 0$ . (SANTOS, 2008, P.59).....	34
FIGURA 13 - ORIENTAÇÃO DA BOIA, COM $V_X = V_Y = 0.5$ . (SANTOS, 2008, P.60).....	34
FIGURA 14 – TRAJETÓRIA DA BOIA COM ALGORITMO <i>WAY POINT</i> . (SANTOS, 2008, P. 60).....	35
FIGURA 15 – TRAJETÓRIA DA BOIA COM O ALGORITMO <i>LINE-TRACKING</i> . (SANTOS, 2008, P. 61).....	36
FIGURA 16 – SATÉLITES GPS. (VERÍSSIMO ET AL, 2008).....	40
FIGURA 17 – ESPECIFICAÇÃO DOS PINOS DO <i>66 CHANNEL LS20031 GPS 10HZ RECEIVER</i> . (SPARKFUN, 2006). ....	45
FIGURA 18 – DIMENSÕES DO <i>66 CHANNEL LS20031 GPS 10HZ RECEIVER</i> . (SPARKFUN, 2006). ....	46
FIGURA 19 – ARDUÍNO MEGA 2560. (ARDUINO, 2006). ....	50
FIGURA 20 – LIGAÇÕES DOS COMPONENTES UTILIZADOS NO ARDUÍNO. ....	53
FIGURA 21 – PROTÓTIPO UTILIZADO PARA TESTAR SISTEMA DE CONTROLO. ....	54
FIGURA 22 – FLUXOGRAMA GERAL DO SISTEMA DE CONTROLO.....	55
FIGURA 23 – FLUXOGRAMA DO BLOCO DE INICIALIZAÇÃO. ....	56
FIGURA 24 – FLUXOGRAMA DA LEITURA DE GPS. ....	57

FIGURA 25 – FLUXOGRAMA DO BLOCO DE NAVEGAÇÃO. ....	59
FIGURA 26 – EFEITO DO CONTROLO PID. ....	60
FIGURA 27 – FLUXOGRAMA DO BLOCO PID/MOTORES. ....	61
FIGURA 28 – <i>PULSE WIDTH MODULATION</i> – PWM. (ARDUINO, 2006).....	63
FIGURA 29 – <i>DUTY CYCLE 25%</i> , VELOCIDADE NO SENTIDO NEGATIVO. (ARDUINO, 2006). ...	63
FIGURA 30 – CONTROLADOR DE MOTORES DE CORRENTE CONTÍNUA. ....	64
FIGURA 31 – MALHA DO SISTEMA DE POSICIONAMENTO DINÂMICO BASEADO EM GPS. ....	64
FIGURA 32 – INTERFACE GRÁFICA DO SIMULADOR EM <i>MATLAB (VERSÃO R2011B)</i> . ....	66
FIGURA 33 – DIAGRAMA DE BLOCOS DO SIMULADOR. ....	67
FIGURA 34 – CENÁRIO DAS SIMULAÇÕES.....	69
FIGURA 35 – TESTE REALIZADO NO DEPARTAMENTO DE ARMAS E ELETRÓNICA NA ESCOLA DE TECNOLOGIAS NAVAIS. (GOOGLE EARTH).....	77
FIGURA 36 – TESTE REALIZADO NA PISCINA DA BASE NAVAL DE LISBOA. (GOOGLE EARTH). .....	78
FIGURA 37 – PRIMEIRO TESTE AO CONTROLADOR PID, REALIZADO EM ÁGUAS PARADAS. (GOOGLE EARTH).....	80
FIGURA 38 – TESTE AO CONTROLADOR PID, APÓS REAJUSTES NA PROGRAMAÇÃO. (GOOGLE EARTH). ....	81
FIGURA 39 – TESTE AO CONTROLADOR PID, APÓS REAJUSTES NAS CONTANTES ( $K_P$ , $K_I$ E $K_D$ ). (GOOGLE EARTH).....	82
FIGURA 40 – TESTE AO CONTROLADOR PID, COM MELHORAMENTO DAS CONTANTES ( $K_P$ , $K_I$ E $K_D$ ). (GOOGLE EARTH).....	83
FIGURA 41 – RIO LIZANDRO, LOCAL UTILIZADO PARA TESTES AO CONTROLADOR PID. ....	84
FIGURA 42 – TESTES NO RIO, A CURTA DISTÂNCIA. (GOOGLE EARTH).....	85
FIGURA 43 – TESTE A LONGAS DISTÂNCIAS, REALIZADO NO RIO LIZANDRO. (GOOGLE EARTH). ....	86
FIGURA 44 – TESTE AO CONTROLADOR PID A LONGA DISTÂNCIA, NO RIO, DEPOIS DE ALTERAÇÕES. (GOOGLE EARTH). ....	87
FIGURA 45 – TESTE AO CONTROLADOR PID, A CURTA DISTÂNCIA. (GOOGLE EARTH).....	88

# LISTA DE TABELAS

---

TABELA 1 – DIVERSOS TIPOS DE MENSAGEM NMEA DO 66 CHANNEL LS20031 GPS 10HZ <i>RECEIVER</i> . (FERREIRA, 2010).....	43
TABELA 2 – DESCRIÇÃO DA MENSAGEM RMC – NMEA. (FERREIRA, 2010).....	44
TABELA 3 – CARACTERÍSTICAS DO RECETOR GPS. (SPARKFUN, 2006). ....	45
TABELA 4 – RELAÇÃO ENTRE MODELO DE ARDUÍNO E O MICROCONTROLADOR UTILIZADO. .	48
TABELA 5 – CARACTERÍSTICAS DO ARDUÍNO MEGA. (ARDUINO, 2006). ....	49
TABELA 6 – COMPONENTES UTILIZADOS NA PLACA ARDUÍNO. ....	52
TABELA 7 – SIMULAÇÃO DO ALGORITMO <i>WAY-POINT</i> . ....	71
TABELA 8 - SIMULAÇÃO DO ALGORITMO <i>LINE-TRACKING</i> . ....	73

# LISTA DE EQUAÇÕES

---

EQUAÇÃO 1 - EQUAÇÃO DE CONTROLO PID. (GUERRA, 2009). .....	26
EQUAÇÃO 2 – TERMO DERIVATIVO. (GUERRA, 2009). .....	30
EQUAÇÃO 3 – CONTROLADOR COM TERMO DERIVATIVO E INTEGRAL. (GUERRA, 2009). .....	31
EQUAÇÃO 4 – ORIENTAÇÃO DE REFERÊNCIA DA BOIA. (SANTOS, 2008). .....	33
EQUAÇÃO 5 – ORIENTAÇÃO DE REFERÊNCIA DA BOIA PELO ALGORITMO <i>LINE-TRACKING</i> . (SANTOS, 2008). .....	37

# LISTA DE ABREVIATURAS

---

et al. (et aliae):	E outros (para pessoas)
n.d.	Não definido
GPS	Sistema de posicionamento global
DGPS	Sistema de posicionamento global diferencial
DP	Posicionamento dinâmico
ROV	Veículo operado remotamente
Wi-Fi	Sem fios
NMEA	National Maritime Electronics Association
PWM	Modulação de largura de impulso
A/D	Analógico/Digital
VHF	Frequência muito alta
UHF	Frequência ultra alta
Tx/Rx	Transmissão/Recepção
GND	Terra, fio negativo
PWR	Energia, fio positivo
FTP	Protocolo de transferência de ficheiros
BNL	Base Naval de Lisboa
PI	Controlador proporcional e integral
PID	Controlador proporcional, integral e derivativo

# CAPÍTULO 1 - INTRODUÇÃO

---

## 1.1. ENQUADRAMENTO

As competições de vela designam-se de regatas. Estas dividem-se em classes dependendo do tamanho do barco e das características dos tripulantes. Para se realizar uma regata é necessário haver um campo de regatas. Este campo é montado e definido por boias, colocadas estrategicamente, não só para definir o trajeto mas também para impor desafios para o trajeto. Tendo em conta a orientação do campo em relação ao vento. Assim, o posicionamento de uma boia é fundamental para o bom funcionamento de uma regata.

As boias que definem o trajeto da regata geralmente são ancoradas em locais pouco profundos (através de uma poita). Deste modo é possível fixá-la ao fundo do mar ou rio. A Figura 1 representa uma boia de regata de forma cilíndrica.



Figura 1 – Boia de regata de forma cilíndrica. (Mundo da Náutica, 2011).

Estas boias incluem quatro pontos de fixação para a linha de ancoragem e para absorverem a força de tensão.

Quando surge a necessidade de colocar estas boias em águas profundas, o sistema de poita já não se torna eficaz devido à distância ao fundo do mar. Neste caso, pode-se utilizar boias de deriva. No entanto não são eficazes para as regatas pois podem afastar-se da posição pretendida.

Outra forma de posicionar uma boia, sem utilizar o sistema de poita, é utilizar um sistema de posicionamento dinâmico. Existem diversos tipos de sistemas de posicionamento dinâmico. Para obter a referência geográfica pode ser através de rede acústica, de

posicionamento baseado em GPS, ou mesmo baseado em visão e laser. A propulsão pode ser através de velas, de motores elétricos ou motores de combustão. (Tannuri, 2009).

## 1.2. JUSTIFICAÇÃO DO TEMA

Este Projeto de Dissertação surge na sequência de contatos realizados junto da Federação Portuguesa de Vela para a construção de um protótipo de boias com *Station Keeping*<sup>1</sup> para regatas. Uma regata consiste na realização de vários percursos, e estes percursos são delimitados principalmente por boias. Para tal, o posicionamento de uma boia é fundamental para o funcionamento de uma regata.

Atualmente os sistemas de posicionamento dinâmico são muito utilizados. Podem-se encontrar estes sistemas de posicionamento em diversas áreas, tais como em plataformas petrolíferas, em navios, e até mesmo em pequenas embarcações que utilizem motores elétricos com *i-pilot*.

Os sistemas de posicionamento dinâmico são importantes para uma área de desenvolvimento da robótica móvel. Existem diversas universidades a desenvolver projetos de robótica móvel, como a Faculdade de Engenharia da Universidade do Porto, que em 2008 participou e organizou a segunda edição da competição mundial de veleiros robóticos. (*World Robotic Sailing Championships*) (Microtransat, *n.d.*).

Na Marinha existe o navio “Almirante Gago Coutinho” que está dotado de um sistema de posicionamento dinâmico que lhe permite operar o ROV a grande profundidade.

## 1.3. OBJETIVOS

A presente dissertação tem como objetivo desenvolver um sistema de posicionamento dinâmico simples e funcional, de modo a solucionar o problema do posicionamento de boias de regata em águas profundas. A aplicabilidade deste sistema numa boia é fulcral para manter as boias sempre na mesma posição. Sem esta aplicação de posicionamento dinâmico as boias podem-se afastar muito da posição pretendida, o que pode levar a que

---

<sup>1</sup> O conceito de *Station Keeping* está associado neste caso a uma boia que mantenha a sua posição de forma autónoma.

seja cancelada a regata. Um caso destes aconteceu em Fevereiro de 2012 na primeira regata do Campeonato da Europa de Vela em que foi cancelado de uma regata devido ao desaparecimento de uma boia. (Lopes, 2012).

Ao longo desta dissertação de mestrados são propostos os seguintes objetivos:

- Fazer uma seleção e realizar um estudo dos equipamentos necessários para criar um sistema de posicionamento autónomo;
- Programar um sistema de posicionamento autónomo com um controlador;
- Testar o sistema de posicionamento autónomo.

#### 1.4. REQUISITOS

Para que o projeto consiga cumprir com os objetivos referidos anteriormente, é necessário estabelecer especificações mínimas requeridas para o seu bom funcionamento. Para isso, são propostos os seguintes requisitos:

- Ser um sistema autónomo;
- Ter capacidade de cálculo;
- Ter rapidez de resposta;
- Posicionamento com condições ambientais aceitáveis para o funcionamento do protótipo.

Ao longo da dissertação serão abordados os seguintes capítulos: Estado de arte, arquitetura do sistema, teste e resultados e conclusão. O capítulo 2, estado de arte, consiste no enquadramento teórico do tema, na definição e seleção dos equipamentos a utilizar, e na descrição dos algoritmos de posicionamento. O capítulo 3, arquitetura do sistema, tem o intuito de descrever as ligações efetuadas dos diversos equipamentos, bem como a programação do mesmo. O capítulo 4, simulador e testes, consiste na simulação dos algoritmos de posicionamento referidos no capítulo 2, e irá abordar todos os testes efetuados bem como os resultados. No capítulo 5 irá constar todas as conclusões desta dissertação.

## CAPÍTULO 2 - ESTADO DE ARTE

---

Este projeto de posicionamento automático é relevante para situações onde não seja possível “fixar” uma boia ao fundo do mar, ou no fundo do leito de um rio. Quando as boias se colocam em locais pouco profundos não surge esta necessidade de posicionamento dinâmico, pois basta colocar um “peso” para fixar ao fundo. No entanto, quando em águas profundas, o mesmo não acontece, e aí surge a necessidade de obter um sistema de posicionamento da boia que contrarie o movimento provocado por qualquer força que a desloque da sua posição (força do vento, estado do mar, etc..).

Para o funcionamento desta boia/plataforma é muito importante saber a posição atualizada. Para isso obtém-se a posição através do Sistema de Posicionamento Global (GPS). Assim, será descrito o funcionamento do GPS, os fatores que afetam o seu funcionamento, quais as suas eventuais aplicações, o Sistema de Posicionamento Global Diferencial (DGPS), e por fim serão obtidos dados que habilitem a escolha do sistema GPS. Este sistema GPS irá ligar a um sistema de comando e controlo que tem como função gerir toda a informação. Serão apresentados alguns tipos de sistema de comando e controlo, bem como o seu *hardware*<sup>2</sup>.

Para finalizar é importante também estudar alguns algoritmos que sejam úteis para manter a posição da boia. De realçar que este sistema tem como base ser um sistema pequeno e de baixo custo.

---

<sup>2</sup> O *hardware* é tudo o que é físico de um equipamento eletrónico.

## 2.1. SISTEMAS DE POSICIONAMENTO DINÂMICO

A motivação responsável pelo aparecimento dos sistemas de posicionamento dinâmico foi relacionada com a exploração de petróleo em águas profundas, de forma a substituir as plataformas fixas por volta da década de 60. O primeiro navio a manter-se posicionado foi o “Cuss-I”, em 1961, nos Estados Unidos. O controlo de posição era efetuado manualmente, ou seja, o operador mantinha a posição através das informações recebidas do sistema radar e do sonar. Ainda nesse mesmo ano, desenvolveu-se o primeiro navio verdadeiramente equipado com sistemas de posicionamento dinâmico, o “Eureka”. (Tannuri, 2009).

Este navio utilizava um controlador analógico que recebia informações de um sensor de posição. Ao longo das décadas 60 e 70, outros navios foram concebidos com sistema de posicionamento dinâmico, como o norte-americano “Cardrill” e o francês “Terébel”. Em comparação com atuais sistemas de posicionamento dinâmico, eram extremamente simples, utilizando controladores analógicos, sem redundância. (Tannuri, 2009).

Sistema de posicionamento dinâmico (*Dynamic Positioning - DP*), tal como o nome indica, significa uma unidade ou um navio com capacidade de se manter autonomamente numa posição (fixar localização ou permanecer na rota) exclusivamente por meio de propulsão ativa.

Para implementar um sistema de posicionamento dinâmico, são requeridas várias unidades, sendo as principais:

- Sistema de posicionamento de referência:

O sistema de posicionamento de referência determina a posição geográfica, podendo obter esta posição de diversas maneiras, como através de sinais acústicos, através de visão e laser, através de recetor GPS, etc..

- Unidade de controlo e monitoração:

A unidade de controlo e monitoração é a área responsável por tratar e converter os dados gerados pelo sistema de posição de referência. O número de controladores também depende do tipo de operação. Esta área tem o dever de transmitir sinais de comando ao

sistema de propulsão de acordo com a sua disponibilidade pela otimização da capacidade operacional do sistema de posicionamento dinâmico.

- Sistema de propulsão:

Este sistema é responsável pela deslocação da embarcação para se posicionar corretamente, recebendo instruções através da unidade de controlo. No caso deste projeto, o sistema irá utilizar um sistema de propulsão constituído por dois motores elétricos.

O sistema de posicionamento dinâmico pode ser absoluto, onde a posição é muito importante, sendo mantida em relação a um ponto fixo em terra, ou relativa a um determinado objeto móvel. Também é utilizado para posicionamento de navios num ângulo favorável em relação à direção dos ventos, correntes e ondulações.

Existem diversas formas de efetuar posicionamento dinâmico, entre os quais através de rede acústica, de posicionamento geo-espacial, ou mesmo baseado em visão e laser. Em seguida são apresentadas as diferentes formas de posicionamento dinâmico, e serão apresentadas, igualmente, as vantagens e desvantagens.

### 2.1.1. POSICIONAMENTO GEOGRÁFICO ATRAVÉS DE REDE ACÚSTICA

O posicionamento geográfico através de rede acústica é utilizado em diversas aplicações comerciais como exploração petrolífera, e científicas como a monitorização ambiental, mapeamento de fundos oceânicos e deteção de objetos.

Este tipo de posicionamento utiliza sinais de rádio debaixo de água para se posicionar. No entanto, a utilização dos sinais rádio debaixo de água torna-se pouco viável, pois estes apenas se propagam a distâncias aceitáveis se forem emitidos a baixas frequências (30Hz-300Hz), necessitando de grandes potências de emissão. (Santos, 2008).

De acordo com Santos (2008), a descoberta da acústica subaquática remonta ao tempo de Leonardo Da Vinci, quando este terá descoberto a possibilidade de ouvir navios à distância através de um tubo emergido no oceano. Apesar destas comunicações serem eficientes e fiáveis, a sua utilização em comunicações subaquáticas revela alguns problemas relacionados com a grande variação das condições acústicas dos meios de operação. Estas variações estão fortemente relacionadas com características oceanográficas como temperatura, salinidade e correntes, que introduzem um elevado grau de incerteza no comportamento dos sinais acústicos emitidos. (Santos, 2008).

As emissões e receções de sinais acústicos surgiram através de simples *transponders*<sup>3</sup>, mas rapidamente evoluíram para módulos de emissão e receção geridos por módulos de controlo, capazes de efetuar simples tarefas de emissão e resposta a deteções de sinais acústicos. Com o evoluir da investigação estes módulos de controlo passaram a integrar um sistema de posicionamento acústico de veículos autónomos submarinos para monitorização ambiental, baseado em faróis acústicos instalados em boias ancoradas no cenário de operação. (Almeida, 2010).

A Figura 2 apresenta um sistema de posicionamento dinâmico através de sono-boias.

---

<sup>3</sup> Conjunto de canais de comunicação que exerce a função de unidade repetidora.

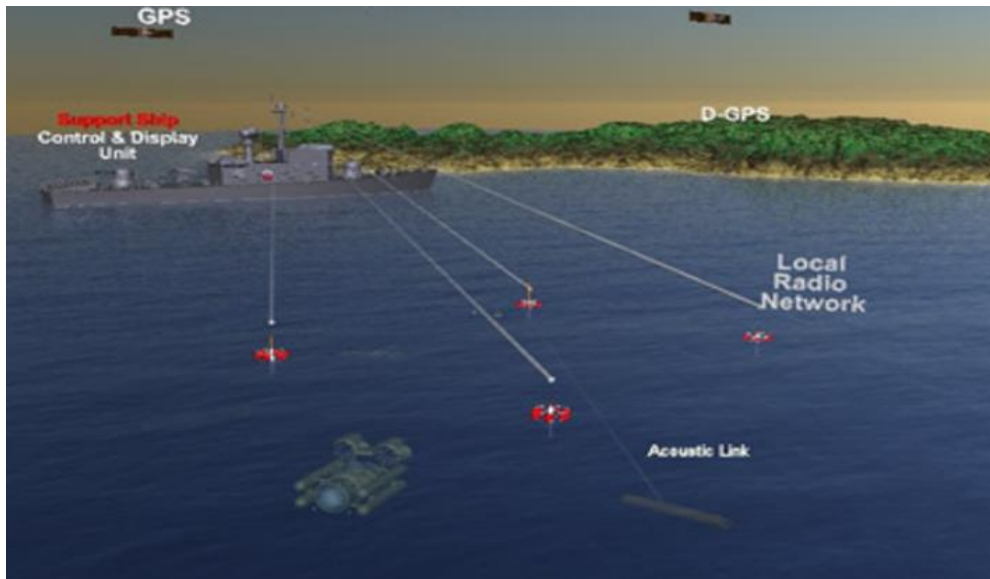


Figura 2 – Sistema de posicionamento dinâmico utilizando sono-boias. (Almeida, 2010).

Este sistema é baseado numa rede de boias de superfície que mede o tempo de chegada de sinais acústicos enviados por um transmissor acústico montado num veículo. Cada boia mede então os tempos de chegada dos dois sinais e envia-os, via rádio, juntamente com a sua posição geográfica para a unidade central de controlo. Assumindo um valor constante para a velocidade de propagação do som na água, a posição do veículo é calculada por triangulação das distâncias às sono-boias. (Almeida, 2010).

### 2.1.2. POSICIONAMENTO GEOGRÁFICO BASEADO EM VISÃO E LASER

Presentemente, a visão computacional é utilizada em muitos trabalhos, sendo o método sensorial viável para aplicação em veículos autónomos ou veículos operados remotamente (*ROV – Remotely Operated Vehicle*). Este método apresenta uma forma precisa de posicionamento do veículo em operações que requerem baixa velocidade, e distância curta entre o veículo e a estrutura a ser inspecionada. (Garcias *et al.*, 2003).

Contudo, a visão computacional possui algumas desvantagens, entre as quais, o facto de ter um campo de visão limitado e quando equipada apenas com uma câmara existe dificuldade em medir distâncias e tamanho dos objetos.

O sistema de visão computacional pode ser emparelhado com um ponteiro de raio laser, possibilitando assim a criação de marcos visuais artificiais, e auxiliando na determinação da distância e na medição do tamanho do objeto alvo. O laser pode ser também utilizado para posicionar embarcações de apoio a plataformas marítimas, conhecidos como *Fanbeam*. (Buscariollo, 2008).

A Figura 3 representa uma embarcação de apoio, utilizando uma *Fanbeam* para se posicionar em relação a uma plataforma fixa.

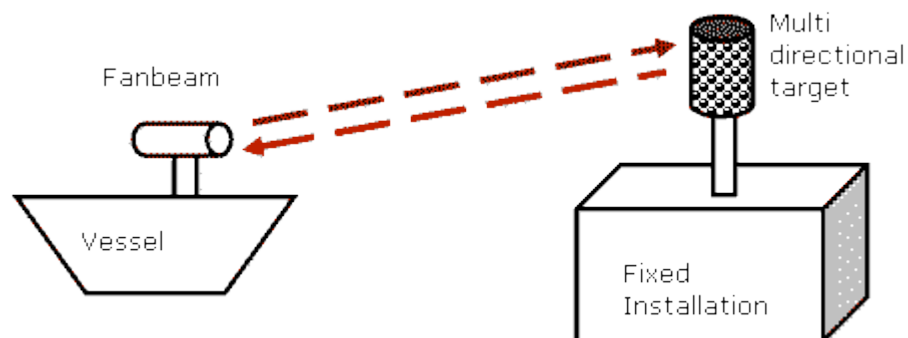


Figura 3 – *Fanbeam*, como sistema de posicionamento dinâmico. (Buscariollo, 2008).

Este sistema de posicionamento dinâmico, utiliza a *Fanbeam* para se posicionar próximo da plataforma, até uma distância 1000 metros com um erro médio de 10 centímetros. Quando atingir a posição desejada, o sistema é realimentado de forma a manter essa posição em relação à plataforma. (Buscariollo, 2008).

Este método sensorial baseado em visão laser é geralmente utilizado também para veículos flutuantes de pequenas dimensões em fins laboratoriais. (Buscariollo, 2008).

### 2.1.3. POSICIONAMENTO GEOGRÁFICO BASEADO EM GPS

O posicionamento dinâmico com sistema de referência baseado em GPS corresponde a um sistema complexo de controlo de posição dinâmica, podendo ser composto por dados de

outras fontes capazes de tornar seu posicionamento mais preciso (*GPS*, *DGPS*, Anemômetros, Giroscópios, Bússolas magnéticas, etc.). (Bray, 1998) e (Fossen, 1994).

O sistema de referência para o *DP* mais comum e o mais utilizado no mercado é o *GPS*. Através do sistema de referência, o *DP* analisa o que está a acontecer e toma as ações necessárias ao posicionamento.

O sistema de controlo de posicionamento dinâmico, recebe sinais dos sistemas de referência, é responsável pela execução do algoritmo de controlo, que indica as correções de posição.

O sistema de propulsão é composto por motores elétricos. Este sistema recebe os dados do sistema de controlo, e atua para se deslocar para a posição.

A Figura 4 apresenta de um modo geral o funcionamento sequencial de um sistema de com posicionamento geográfico baseado em GPS.



Figura 4 – Sistema de posicionamento geográfico baseado em GPS.

O sistema inicia com a receção do posicionamento geográfico da boia. O algoritmo de controlo analisa a posição geográfica recebido com a desejada e vai atuar nos propulsores.

#### 2.1.4. ESCOLHA DO SISTEMA DE POSICIONAMENTO DINÂMICO

Tendo em conta os equipamentos que a Escola Naval possui, o sistema de posicionamento dinâmico escolhido é o posicionamento geográfico baseado num recetor GPS. Devido ao projeto “*eVentos*”, a Escola Naval já possui *hardware* indicado para este sistema de posicionamento dinâmico. No subcapítulo *HARDWARE* é feito um estudo e descrito o *hardware* que se vai utilizar.

Na constituição deste sistema, apenas o sistema de referência é idêntico ao do projeto “*eVentos*”. Isto porque para ambos se utiliza o sistema de posicionamento geo-espacial, baseado em GPS. Ao contrário do projeto “*eVentos*”, que é um veleiro e utiliza a força do vento para navegar, este projeto não possui nenhum sistema de velas e então necessita de um sistema de propulsão constituído por dois motores elétricos. Como sistema de direção o projeto “*eVentos*” utiliza um leme, no entanto para o projeto desta dissertação o sistema de direção é efetuado através do diferencial dos dois motores. Assim, o sistema de controlo de posicionamento obrigatoriamente tem que ser diferente.

Após escolher o sistema de posicionamento dinâmico, surge a necessidade de saber que tipo de controlo é que se vai utilizar neste projeto. De seguida, é realizado um estudo sobre algoritmos de controlo de posição.

## 2.2. SISTEMA DE CONTROLO PID

O sistema de controlo Proporcional-Integral-Derivativo (PID) é o controlador mais usado na indústria e tem sido utilizado em todo o mundo para sistemas de controlo industrial. (OGATA, 2002).

A popularidade de controladores PID é atribuída em parte ao seu desempenho robusto numa ampla gama de condições de funcionamento e em parte à sua simplicidade funcional, que permite aos engenheiros operá-los de uma forma simples e direta. (OGATA, 2002).

Como o nome indica, o algoritmo PID é composto por três coeficientes: proporcional, integral e derivativo, que são variados para obter a resposta ideal. Neste capítulo será estudado sistemas de circuito fechado, a teoria do PID e os efeitos do ajuste de um sistema de controlo em malha fechada. (OGATA, 2002).

### 2.2.1. EQUAÇÃO PID

A Equação 1 representa a equação de controlo PID.

$$y(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt}$$

Equação 1 - Equação de controlo PID. (Guerra, 2009).

Onde:

$K_p$  = constante proporcional;

$K_i$  = constante integral;

$K_d$  = constante derivativa.

- **$K_p$  - Correção proporcional ao ERRO:**

A correção a ser aplicada ao processo deve crescer na proporção que cresce o erro (entre o valor real e desejado).

- **Ki - Correção proporcional ao integral do ERRO:**

Esta correção é o somatório do tempo em que o erro persiste. Erros pequenos mas que existem há muito tempo requerem correção mais intensa.

- **Kd - Correção proporcional à taxa de variação do ERRO:**

Se o erro está a variar muito rápido, esta taxa de variação deve ser reduzida para evitar oscilações.

Para o projeto se descolar para a posição desejada, é necessário saber o erro. Este erro é a diferença entre a proa do projeto e o azimuth da posição desejada. A parte integral é a soma dos valores de erro anterior. A secção derivativa representa a taxa de variação do próprio erro. (Palma, 2008).

O  $K_p$  coeficiente proporcional, o  $K_i$  coeficiente integral, e o  $K_d$  coeficiente derivativo são coeficientes de ganho que sintonizam o controlador PID. O  $y(t)$  é o esforço total de controlo aplicado aos atuadores (motores) para alcançar e manter o ponto desejado. (Palma, 2008).

## 2.2.2. CONTROLO EM MALHA FECHADA

*“The presence of disturbances is one of the main reasons for using control. Without disturbances there is no need for feedback control” (Aström, 1997).*

A Figura 5 representa o sistema de controlo PID em malha fechada.

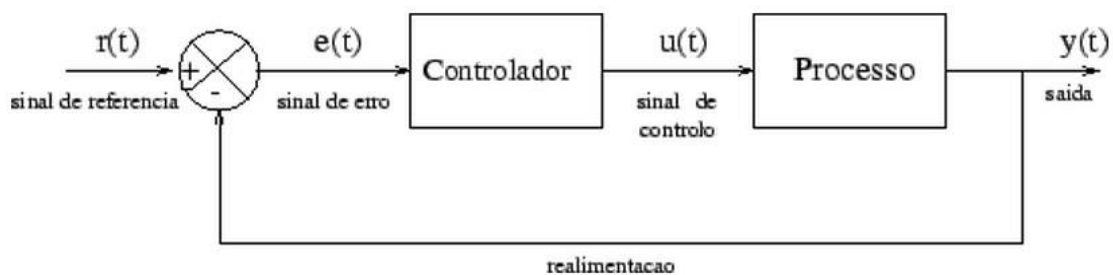


Figura 5 - Controlo em malha fechada. (Silva, 2000).

No controlo em malha fechada, as informações sobre como a saída de controlo está a evoluir são utilizadas para determinar o sinal de controlo que deve ser aplicado ao *processo* no instante seguinte. Isto é feito a partir de uma *realimentação* da saída para a entrada. (Silva, 2000).

Em suma, a fim de tornar o sistema mais preciso e de fazer com que ele reaja a perturbações externas, o sinal de saída é comparado com o sinal de referência (*set-point*) e com o desvio (erro). Entre estes dois sinais é determinado o sinal de controlo que deve efetivamente ser aplicado ao processo. Assim, o sinal de controlo é determinado de forma a corrigir o desvio entre a saída e o sinal de referência. O dispositivo que determina o sinal de controlo através do sinal de erro e do sinal de referência é o *controlador ou compensador*. O diagrama básico de um sistema de controlo em malha fechada é mostrado na Figura 5. (Silva, 2000).

Nesta metodologia usual procura-se que o *Controlador* tenha a estrutura mais simples possível, para que o método de cálculo fique facilitado e o número de parâmetros a ajustar seja mínimo. De outro modo, a afinação inicial e eventuais refinamentos tornam-se pouco praticáveis. (Palma, 2008).

Com um número restrito de parâmetros (P, PI, PD ou PID) não é possível, geralmente, fazer a compensação plena da dinâmica do sistema que se pretende controlar. Assim, procura-se que o sistema resultante em cadeia fechada tenha um desempenho satisfatório, e que tenha as seguintes características:

- Ser estável com margem suficiente;
- Ter um erro admissível na resposta;
- Ter a rapidez necessária;
- Ter a evolução transitória da resposta com as características desejadas. (Palma, 2008).

### 2.2.3. RESPOSTA PID

- CONTROLO PROPORCIONAL – P

O facto de ter simplesmente controlo proporcional pode ter algumas desvantagens. A principal desvantagem deste controlador é o aumento de *overshoot* à medida que a constante proporcional aumenta, como ilustrado na Figura 6. Outra desvantagem deste controlador é o facto de uma mudança abrupta na variável de processo pode causar um pico na saída do controlador muito acima do valor desejado. (Guerra, 2009).

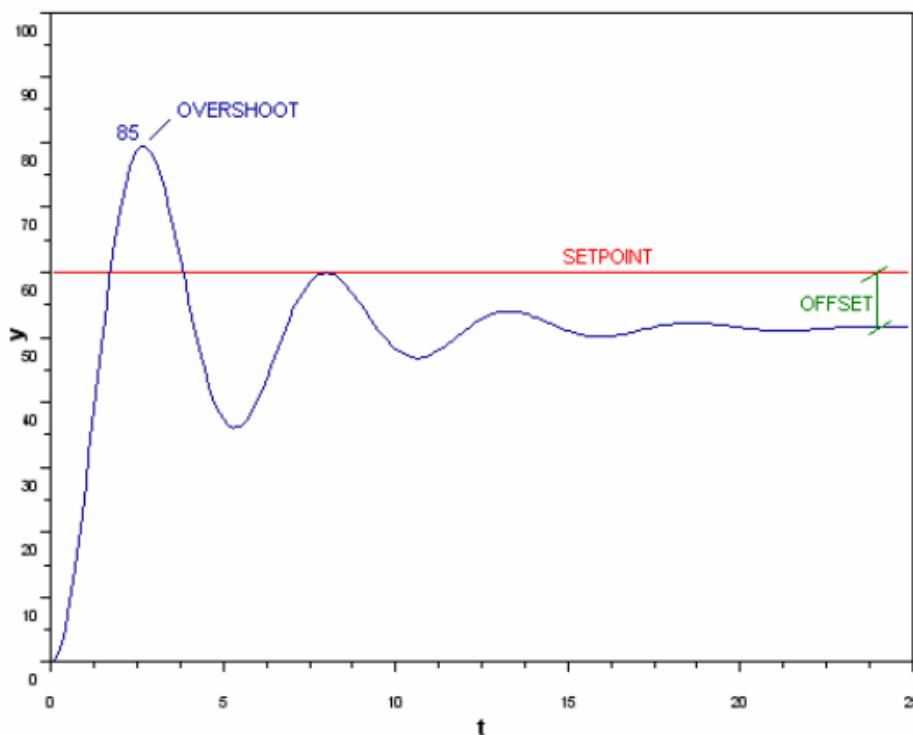


Figura 6 – Controlador proporcional. (Guerra, 2009).

- CONTROLO PROPORCIONAL INCLUINDO CONTROLO DERIVATIVO – PD

De forma a eliminar o problema do *overshoot* do controlador proporcional é introduzido um termo derivativo. Este termo corresponde à taxa de mudança de erro, ilustrado na Equação 2. Assim, o controlador irá ter uma resposta inicial mais elevada e diretamente relacionada à taxa de mudança de erro. Quanto maior a taxa de mudança de erro, mais rápida é a resposta do controlador à mudança. (Guerra, 2009).

$$G_c(s) = K_p + K_d \cdot s$$

Equação 2 – Termo derivativo. (Guerra, 2009).

Uma desvantagem neste tipo de controlo é a existência de *offset* (erro permanente quando o sistema estabiliza), ilustrado na Figura 7. Este erro deve-se ao facto de quando o *desvio* se torna nulo, a saída do controlador também se anula. Assim o sistema tende a equilibrar-se em torno do *SetPoint*. (Guerra, 2009).

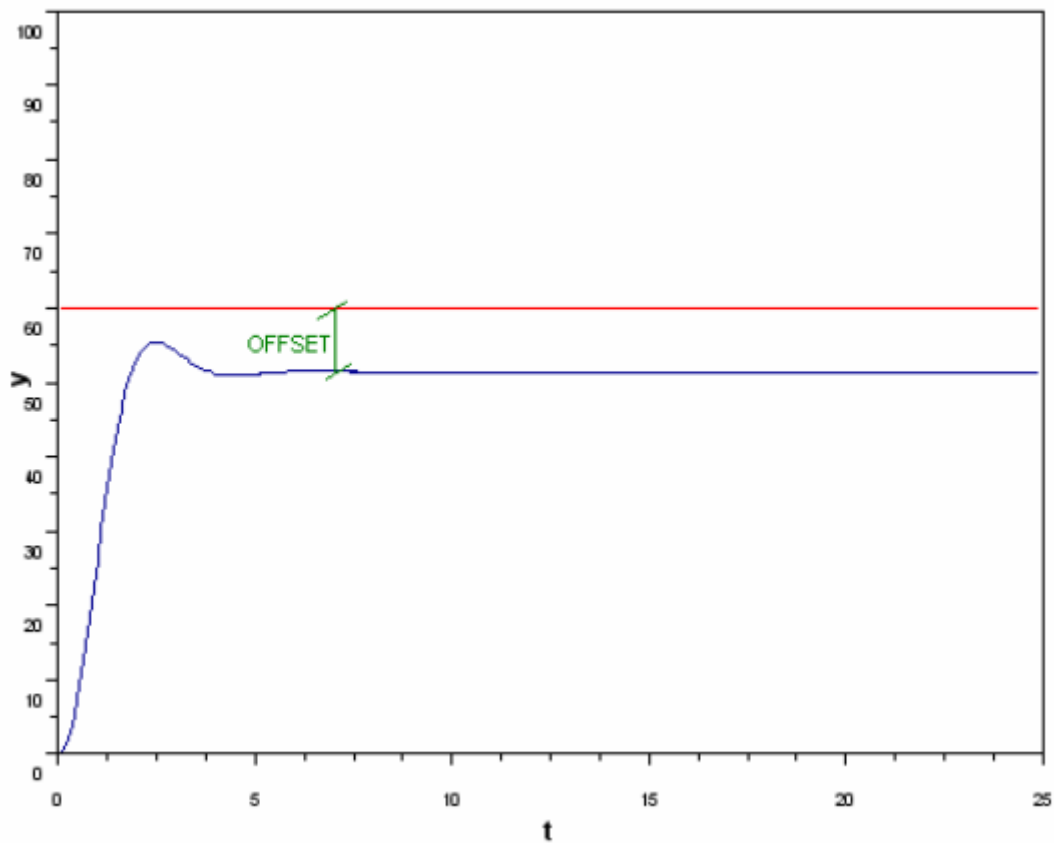


Figura 7 – Controlador proporcional e derivativo. (Guerra, 2009).

- CONTROLO PROPORCIONAL, INTEGRAL E DERIVATIVO – PID

Uma forma de anular o *offset* é adicionar um termo integral ao controlador (ver Equação 3).

$$G_c(s) = K_p + K_d \cdot s + \frac{K_i}{s}$$

Equação 3 – Controlador com termo derivativo e integral. (Guerra, 2009).

Com este termo adicionado na Equação 3, em relação à Equação 2 uma entrada nula (*desvio* nulo), não resulta numa saída nula do controlador. Deste modo, tem a desvantagem de passar a ter uma resposta transitória mais lenta – Figura 8. (Guerra, 2009).

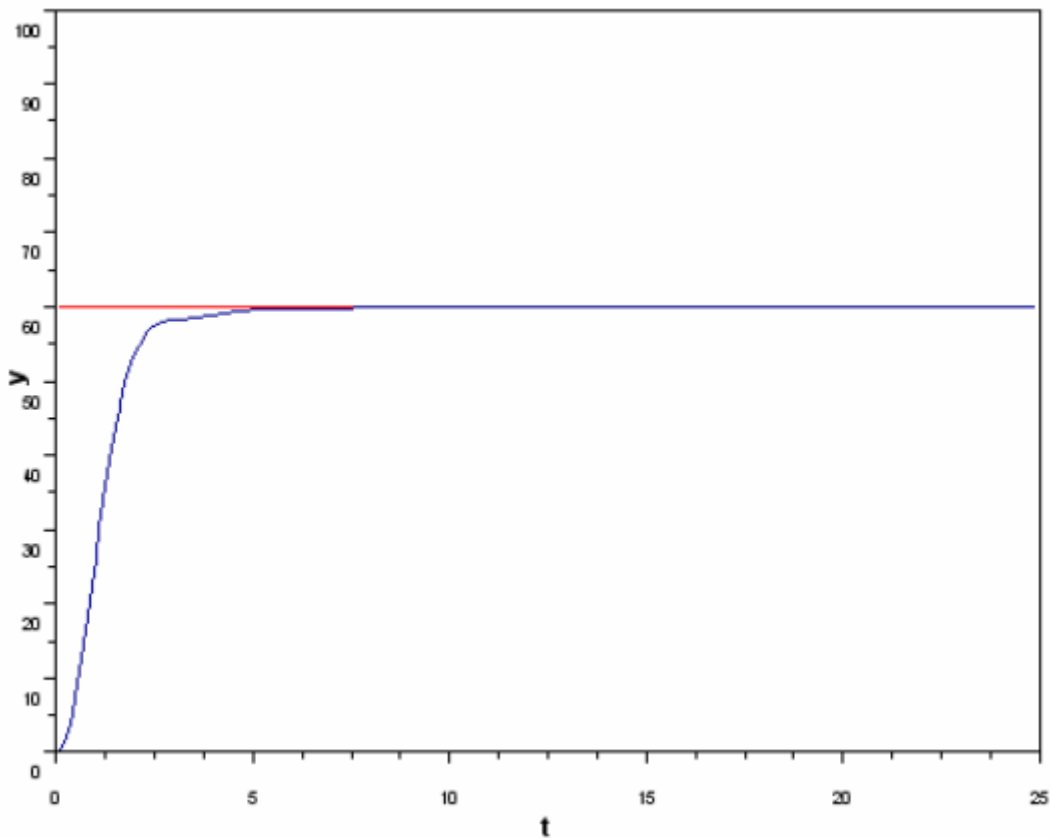


Figura 8 – Controlador proporcional, integral e derivativo. (Guerra, 2009).

#### 2.2.4. LIMITADORES

A aplicação de limitadores destina-se, usualmente, a fins de proteção (por exemplo, correntes máximas) ou a restrição de condições (amplitudes máximas de modulação de largura de impulso). Estes limitadores vão limitar a saída do controlador de modo a suspender a ação integral sempre que a limitação intervém. Como, durante esta fase, a saída não segue a referência, o erro será acumulado pela ação integral. Para recuperar é necessário receber idêntico integral de erro em sentido contrário, como mostra na Figura 9. (Palma, 2008).

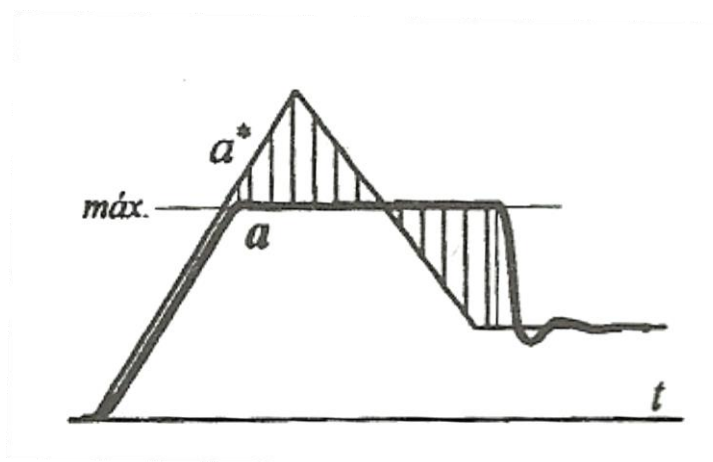


Figura 9 - Resposta do controlador sem suspender a ação integral. (Palma, 2008).

Denomina-se este fenómeno de *wind up* ou “colagem” do integrador. Se a implementação for apropriada para suprimir a ação integral enquanto decorre a limitação, a resposta retomará imediatamente a evolução desejada quando a saturação terminar. Enquanto durar a intervenção dum limitador, o comportamento do sistema não é o mesmo que se prevê sem qualquer restrição. (Palma, 2008).

#### 2.2.5. OUTROS ALGORITMOS

A boia quando posicionada no mar está sujeita a perturbações induzidas pelas condições atmosféricas (correntes, ventos...) afetando o seu posicionamento e consequentemente aumentando o erro de posição.

Em seguida são apresentados os dois algoritmos de posicionamento da boia que mais se adequam para o projeto, e serão apresentadas, igualmente, as vantagens e desvantagens.

▪ **ALGORITMO WAY-POINT**

O algoritmo *way-point* consiste em calcular, a cada instante, a orientação de referência da posição atual para a posição desejada apresentada na Equação 4. Resulta num “apontar”, em cada instante, para a posição pretendida.

$$\psi_{ref} = \arctan\left(\frac{x_d - x}{y_d - y}\right)$$

Equação 4 – Orientação de referência da boia. (Santos, 2008).

Onde:

$\psi_{ref}$  - orientação de referência

$y_d, x_d$  - posição desejada

$y, x$  - posição atual

Analisando este algoritmo e assumindo um valor normalizado de 1 para a velocidade, obtém-se as seguintes figuras que correspondem à orientação instantânea da boia, face a perturbações ( $v_x$  e  $v_y$ ).

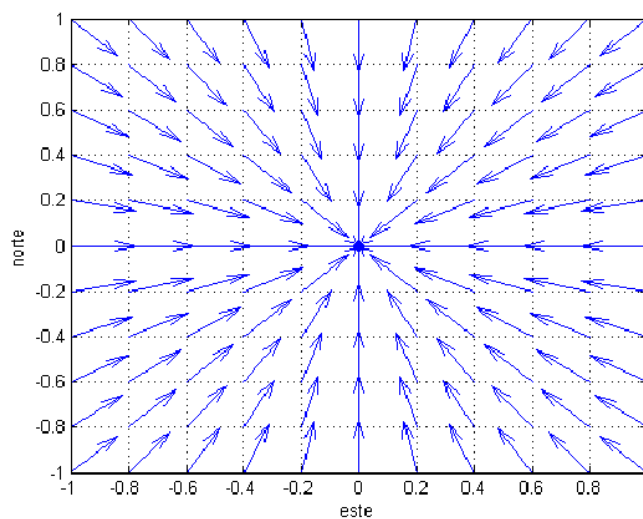


Figura 10 - Orientação da boia, com  $v_x = v_y = 0$ . (Santos, 2008, p.59).

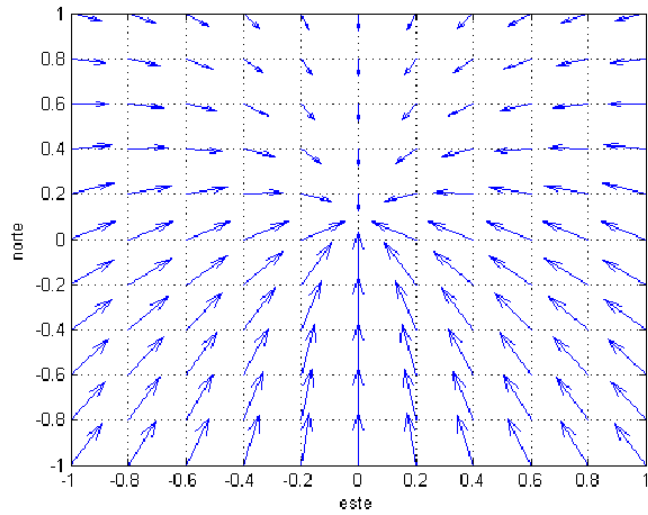


Figura 11 - Orientação da boia, com  $v_x = 0$ ;  $v_y = 0.5$ . (Santos, 2008, p.59).

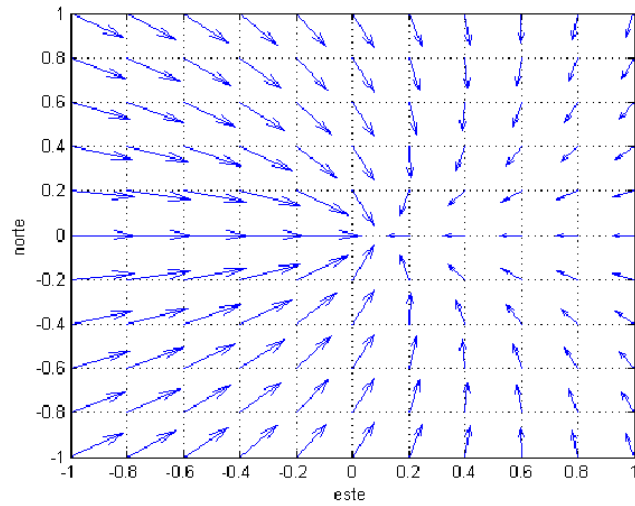


Figura 12 - Orientação da boia, com  $v_x = 0.5$ ;  $v_y = 0$ . (Santos, 2008, p.59).

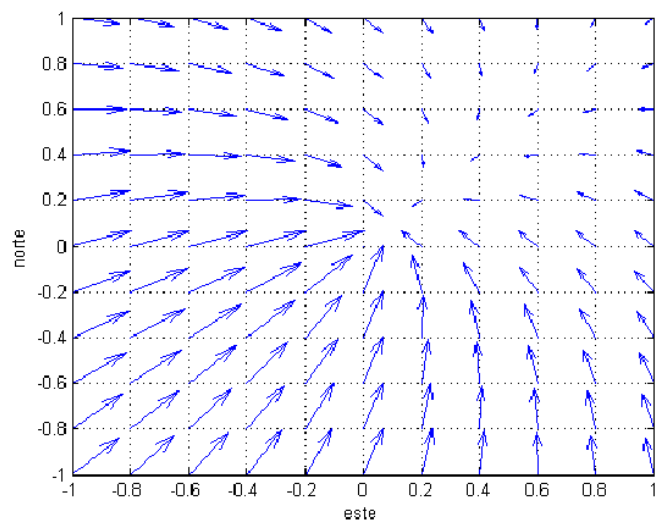


Figura 13 - Orientação da boia, com  $v_x = v_y = 0.5$ . (Santos, 2008, p.60).

Analisando as orientações, verifica-se que as trajetórias serão curvilíneas, exceto quando  $v_x = v_y = 0$  como se pode observar na Figura 10. Quando existem perturbações em  $v_x$  ou  $v_y$ , ou ainda nas duas componentes, as trajetórias apenas são retilíneas quando a boia se desloca no sentido direto ou inverso ao vetor das perturbações. (Santos, 2008).

Quando se utiliza o algoritmo *way-point* a boia não irá seguir a trajetória desejada quando existirem perturbações. A Figura 14 exemplifica uma trajetória curvilínea que a boia iria executar com o algoritmo *way-point* face a uma perturbação perpendicular à sua velocidade.

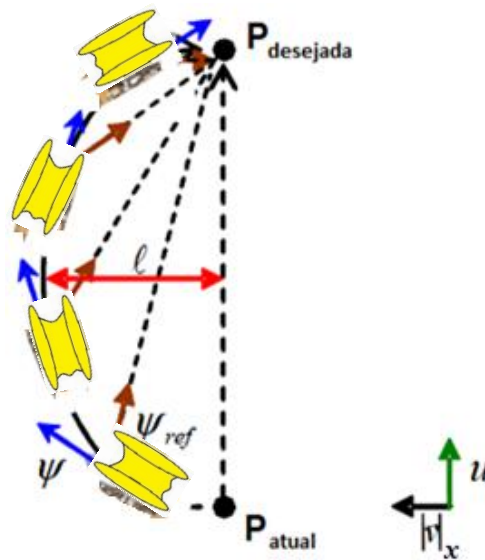


Figura 14 – Trajetória da boia com algoritmo *way point*. (Santos, 2008, p. 60).

Legenda:

$u$  - vetor da velocidade da boia;

$\vec{v}_x$  - vetor da velocidade das perturbações em x;

$\psi$  - orientação real;

$\psi_{ref}$  - orientação de referência (*way-point*).

Na figura anterior o valor de  $\ell$  aumentará quanto maior a dimensão de  $|\vec{v}|_x$  face a  $|\vec{v}|_y$ . A presente técnica deve ser utilizada quando as perturbações perpendiculares ao deslocamento da boia, no caso da figura anterior  $|\vec{v}|_x$ , for nula ou reduzida. Caso o valor das perturbações  $|\vec{v}|_x$  seja elevado, face a  $|\vec{v}|_y$ , a boia divergirá e dificilmente conseguirá atingir o ponto desejado, pois este algoritmo não apresenta qualquer compensação a perturbações. (Santos, 2008).

#### ▪ ALGORITMO *LINE-TRACKING*

O algoritmo *line-tracking* tem como princípio minimizar o erro entre a trajetória real e a desejada. Para casos em que é necessário seguir uma dada trajetória, independentemente das perturbações externas, o método analisado anteriormente não é suficiente.

O algoritmo de controlo *line-tracking* projeta a sua posição atual na linha da trajetória desejada,  $P_i$ , e ainda um outro ponto na mesma linha  $d$  metros à frente,  $P_{i+1}$ , ao contrário do método *way-point*, que apontava sempre para o ponto desejado. Apontando para esse ponto consegue-se corrigir a orientação da boia de forma a compensar perturbações, ficando com erro constante ao longo da manobra. (Santos, 2008).

A Figura 15 demonstra os parâmetros envolvidos no presente algoritmo.

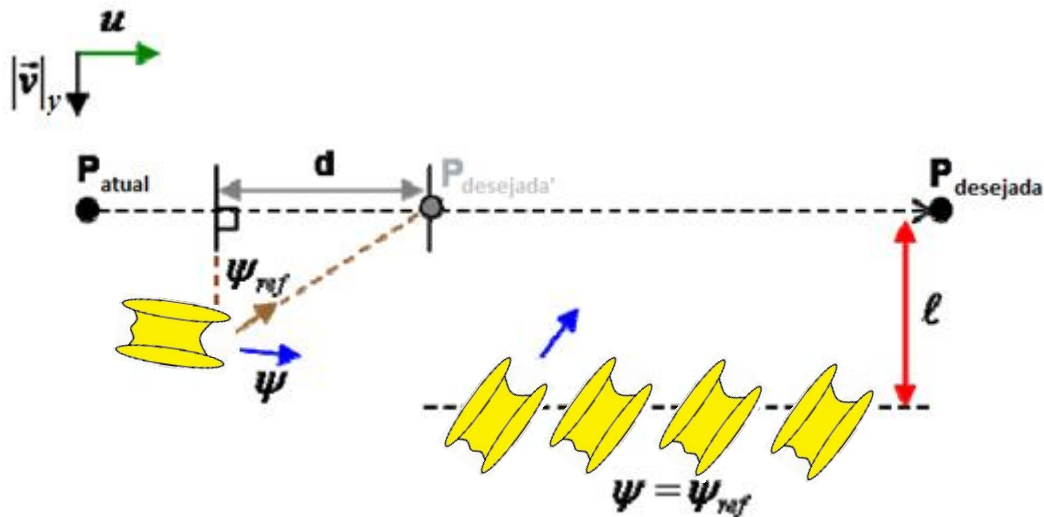


Figura 15 – Trajetória da boia com o algoritmo *line-tracking*. (Santos, 2008, p. 61).

Legenda:

$u$  - vetor da velocidade longitudinal da boia;

$|\vec{v}|_y$  - vetor da velocidade das perturbações em y;

$\psi_{ref}$  - orientação de referência, calculada segundo o método *line-tracking*.

O ângulo da orientação de referência, calculada pelo método *line-tracking* é dado pela seguinte equação:

$$\psi_{ref} = \arctan \left( \frac{x_{P_{i+1}} - x}{y_{P_{i+1}} - y} \right)$$

Equação 5 – Orientação de referência da boia pelo algoritmo *line-tracking*. (Santos, 2008).

Onde:

$\psi_{ref}$  - orientação de referência

$y_{P_{i+1}}, x_{P_{i+1}}$  - posição no instante seguinte

$y, x$  - posição atual

A principal vantagem deste algoritmo prende-se com o facto de se ter um só parâmetro para configurar,  $d$  (distância entre a posição atual e a posição no instante seguinte), o que permite suavizar ou aumentar o ângulo de “ataque” para aproximação à linha, e também a compensação das perturbações.

A principal desvantagem é o erro, sempre existente, no regime permanente,  $\ell$ , que resulta numa trajetória real paralela à desejada.

### 2.3. *HARDWARE*

O sistema de posicionamento deste projeto vai utilizar como sistema de referência o posicionamento geográfico baseado num recetor GPS. Deste modo será efetuado um estudo sobre GPS e análise do GPS a utilizar.

Para determinar qual o sistema de controlo de sistema posicionamento dinâmico vai-se realizar um estudo acerca de sistemas de comando e controlo.

Este subcapítulo apresenta e estuda os equipamentos a serem utilizados neste sistema de posicionamento. Estes equipamentos englobam o posicionamento e o sistema de comando e controlo.

#### 2.3.1. *GLOBAL POSITIONING SYSTEM*

O sistema GPS (*Global Positioning System*) concebido pelo Departamento de Defesa dos EUA no início da década de 1960, sob o nome de 'projeto NAVSTAR' disponibiliza a capacidade de localizar qualquer pessoa no planeta. O sistema foi declarado totalmente operacional apenas em 1995. (Gopi, 2005).

Para indicar a posição de um recetor num lugar à superfície da Terra, utilizam-se coordenadas geográficas: latitude, longitude e altitude. (Veríssimo, Ferreira, Serrasqueiro & Pereira, 2008).

### FUNCIONAMENTO

O sistema GPS consiste em 24 satélites que orbitam a terra a 20.200 km. Esta constelação de satélites, em cada dia, faz duas órbitas e emitem simultaneamente sinais de rádio codificados. Cada satélite emite um sinal que contém: código de precisão (P); código geral (CA) e informação de estado.

Como em outros sistemas de radionavegação, todos os satélites enviam seus sinais de rádio exatamente ao mesmo tempo, permitindo ao recetor avaliar o intervalo de tempo entre emissão/recepção.

Os recetores GPS em qualquer parte do mundo mostram a mesma hora, minuto, segundo, até ao milissegundo. A hora-padrão é altamente precisa, uma vez que cada satélite tem um relógio atómico, com precisão de nano segundo - mais preciso que a própria rotação da Terra. (Gopi, 2005).

O recetor tem que reconhecer as localizações dos satélites. Uma lista de posições, conhecida como almanaque, é transmitida de cada satélite para os recetores. Na terra as estações de controlo rastreiam os satélites e mantêm seus almanaques atualizados. Cada satélite tem códigos P e CA únicos, e o recetor pode distingui-los. (Gopi, 2005).

O GPS é um sistema de simples utilização e acessível a qualquer utilizador. Cada satélite GPS envia sinais com poucas informações e de baixa potência. Os sinais GPS são fracos, pelo que se podem confundir com o ruído de fundo (de rádio) inerente à Terra. Assim os dados de almanaque permitem ao recetor identificar qual o satélite que está a transmitir, a qualquer momento do dia, e discriminar o ruído de sinal válido. Com um mínimo de três satélites, o recetor pode determinar uma posição Latitude/Longitude - que é chamada posição fixa 2D - bidimensional (deve-se entrar com o valor aproximado da altitude para melhorar a precisão). Com a recepção de quatro ou mais satélites, um recetor pode determinar uma posição 3D - tridimensional, isto é, Latitude/Longitude/Altitude. Pelo processamento contínuo de sua posição, um recetor pode também determinar velocidade e direção do deslocamento. (Gopi, 2005).

O funcionamento do sistema GPS pode ser afetado por diversos fatores. Um destes fatores é a precisão da 'Geometria dos Satélites' - localização dos satélites em relação uns aos outros sob a perspetiva do recetor GPS. Se um recetor GPS estiver localizado sob 4 satélites e todos estiverem na mesma região do céu, sua geometria é pobre. Na verdade, o recetor pode não ser capaz de se localizar, pois todas as medidas de distância provêm da mesma direção geral. Isto significa que a triangulação é pobre e a área comum da

---

intersecção das medidas é muito grande (ou seja, a área onde o recetor retira a sua posição cobre um grande espaço). Dessa forma, mesmo que o recetor mostre uma posição, a precisão não é boa. Com os mesmos 4 satélites, se espalhados em todas as direções, a precisão melhora drasticamente. Suponhamos os 4 satélites separados em intervalos de 90° a norte, sul, leste e oeste. A geometria é ótima, pois as medidas provêm de várias direções. A área comum de intersecção é muito menor e a precisão muito maior. (Gopi, 2005).

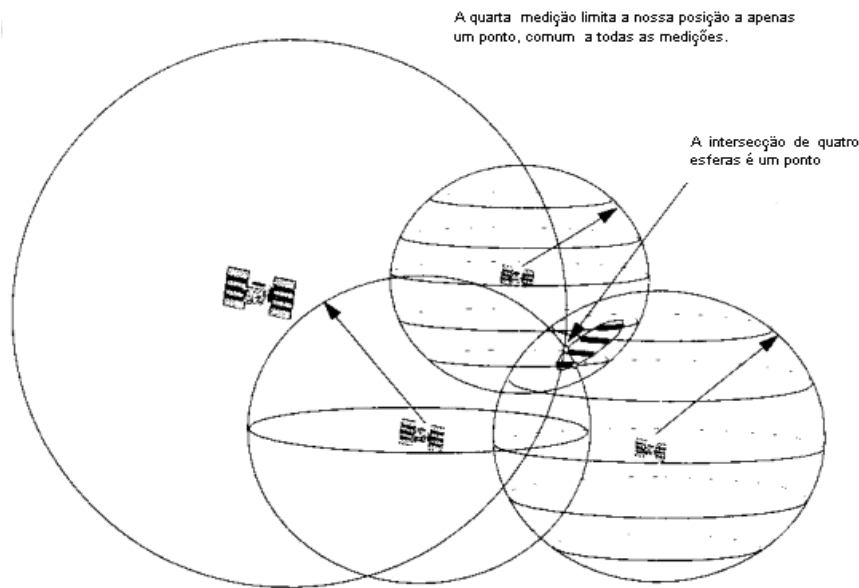


Figura 16 – Satélites GPS. (Veríssimo et al, 2008).

Outro fator de que influência o desempenho do GPS é a interferência resultante da reflexão do sinal em algum objeto. Como o sinal leva mais tempo para alcançar o recetor, este 'entende' que o satélite está mais longe que na realidade.

O atraso na propagação dos sinais deve-se aos efeitos atmosféricos e a alterações do relógio interno. Em ambos os casos, o recetor GPS é projetado para compensar os efeitos.

Existem também erros no recetor, tais como o efeito de ruído do recetor e os erros devido ao multicaminho. Multicaminho é um erro que pode ocorrer na última fase da deslocação do sinal de GPS entre o satélite e o recetor, uma vez que os sinais podem refletir-se em obstáculos locais antes de chegarem ao recetor. Deste modo, o sinal demora mais tempo a chegar ao recetor do que se tivesse deslocado num percurso direto e o recetor assume que o satélite está mais longe do que na verdade está.

Os erros na medição da distância ao satélite são afetados pelo *ruído branco*<sup>4</sup> e pelos efeitos das interferências. (Gopi, 2005).

## SISTEMA DE POSICIONAMENTO GLOBAL DIFERENCIAL

O DGPS é um sistema que permite ao utilizador obter uma precisão que pode variar entre 2 cm a 5 m, através do processamento contínuo de correções nos sinais. O funcionamento do DGPS baseia-se na suposição de que o erro na determinação de um ponto é semelhante para todos os recetores situados num raio até centenas de quilómetros. A eliminação desse erro permite que o utilizador reduza significativamente o erro total presente. Neste sistema, um recetor GPS base é instalado num local georreferenciado. Isto permite conhecer a diferença entre a posição obtida através do recetor de GPS base e a posição real do equipamento, possibilitando a correção diferencial do erro. (Kaplan & Hegarty, 2006).

## APLICAÇÕES

Um recetor GPS pode ter várias aplicações, como por exemplo:

- Mapa dinâmico: o recetor envia a posição para um computador portátil que a visualiza através de um ícone sobre um mapa da região;
- Piloto automático: o recetor alimenta continuamente um piloto automático com dados atualizados, que os utiliza para ajustar a direção e permanecer no rumo. (Gopi, 2005).

O recetor deve usar uma linguagem que o equipamento a ele associado possa entender. Existe um protocolo de comunicação padrão para equipamentos de navegação, Protocolo NMEA - *National Maritime Electronics Association*. Este protocolo é utilizado para comunicação entre sistemas. Existem diferentes versões de protocolos NMEA. Os mais comuns são: 180; 182; 183 versão 1,5; 183 versão 2,0. A maioria dos recetores tem saída NMEA de dados. (Gopi, 2005).

---

<sup>4</sup> Ruído branco é um tipo de ruído produzido pela combinação simultânea de sons de todas as frequências.

## COMUNICAÇÃO - PROTOCOLO NMEA

O protocolo NMEA é um conjunto de especificações de formato de dados e representações elétricas para comunicação de dispositivos eletrônicos de navegação tais como anemômetros, giro bússolas, pilotos automáticos, recetores GPS e muitos outros tipos de instrumentos. Neste projeto, o protocolo NMEA é utilizado para comunicação entre o recetor GPS e o sistema de comando e controlo. Assim, é realizado um estudo de como o GPS comunica com o protocolo NMEA.

Regras do Protocolo:

- Cada início de mensagem começa com o símbolo cifrão - \$.
- Os próximos 5 caracteres indicam a origem da mensagem, sendo dois para o equipamento de origem e três para o tipo de mensagem.
- Todos os campos dos dados são delimitados por vírgulas.
- Quando não há dados disponíveis o campo recebe um Byte nulo.
- O primeiro carácter do último campo deve ser um asterisco - \*, porém isto só se aplica quando houver *checksum*<sup>5</sup>.
- O asterisco é seguido de um *checksum* de dois dígitos representados na forma hexadecimal.
- A mensagem é terminada com um código de Nova Linha - <CR>, ou <LF>, ou "\n".

No recetor de GPS escolhido, *66 Channel LS20031 GPS 10Hz Receiver*, existem alguns tipos de mensagem NMEA que se pode utilizar. A Tabela 1 indica os tipos de mensagem NMEA que este recetor possui.

---

<sup>5</sup> *Checksum* ou soma de verificação é um código usado para verificar a integridade de dados transmitidos.

Mensagem NMEA	Descrição
GGA	Time, position and fix type data.
GLL	Geographic position - latitude/longitude
GSA	GNSS DOP and active satellites
GSV	GNSS satellites in view
RMC	Recommended minimum specific GNSS data
VTG	Course over ground and ground speed

Tabela 1 – Diversos tipos de mensagem NMEA do *66 Channel LS20031 GPS 10Hz Receiver*. (Ferreira, 2010).

Especificações das mensagens:

Ao nível das mensagens transmitidas este protocolo define vários tipos de mensagens (GGA; GLL; GSA; GSV; RMC; VTG) sendo que neste trabalho apenas se utiliza as mensagens do tipo GPR, já que estas fornecem toda a informação necessária ao projeto.

Na Tabela 2 pode-se observar a descrição da mensagem RMC – NMEA.

As mensagens do tipo RMC estão organizadas no formato seguinte:

\$GPRMC,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,<10>,<11>,<12>\*<13><CR><LF>

Exemplo:

\$GPRMC,104549.04,A,2447.2038,N,12100.4990,E,016.0,221.0,250304,003.3,W,A\*22<CR><LF>

Número	Exemplo	Descrição
1	104549.04	Tempo no formato hhmmss.ss, 000000.00 ~ 235959.99
2	A	Estado, 'V' = navigation receiver warning, 'A' = valid position
3	2447.2038	Latitude no formato dddmm.mmmm
4	N	Latitude hemisphere indicator, 'N' = North, 'S' = South
5	12100.4990	Longitude no formato dddmm.mmmm
6	E	Indicador do hemisfério da latitude, 'E' = East, 'W' = West
7	016.0	Velocidade no solo, 000.0 ~ 999.9 nós
8	221.0	Direção no solo, 000.0 ~ 359.9 graus
9	250304	Data UTC, format ddmmyy
10	003.3	Varição magnética, 000.0 ~ 180.0 graus
11	W	Direção da variação magnética, 'E' = East, 'W' = West
		Mais indicadores
		'N' = Data invalid; 'A' = Autonomous; 'D' = Differential; 'E' =
12	A	Estimated
13	22	<i>Checksum</i>

Tabela 2 – Descrição da mensagem RMC – NMEA. (Ferreira, 2010).

### RECETOR GPS 66 CHANNEL LS20031

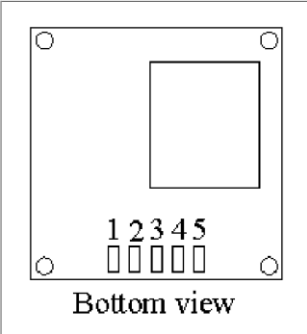
De acordo com Rocha (2011) e Marques (2012), o recetor GPS a utilizar será o *66 Channel LS20031 GPS 10Hz Receiver* da *Sparkfun*, que faz parte do sistema de posicionamento dinâmico utilizado por estes autores nas suas teses de mestrado pela Escola Naval. E é também o disponível pela Escola Naval.

A Tabela 3 indica as características do recetor GPS a utilizar.

<b>Fornecedor</b>	Sparkfun
<b>Nome</b>	LS20031 GPS
<b>Nº Canais</b>	66
<b>Update Rate</b>	10Hz
<b>DGPS</b>	Sim
<b>Interface</b>	NMEA
<b>Antena</b>	Cerâmica, Onboard
<b>Hot Start</b>	2s
<b>Precisão</b>	2,5-3m
<b>Preço</b>	46 €

Tabela 3 – Características do recetor GPS. (Sparkfun, 2006).

Na Figura 17 e Figura 18, encontram-se algumas das especificações deste recetor, bem como as suas dimensões e a especificação dos pinos.



Bottom view

**LS20031**

Pin #	Name	Type	Description
1	VCC	P	Power input
2	RX	I	Data input (TTL level)
3	TX	O	Data output (TTL level)
4	GND	P	Ground
5	GND	P	Ground

Figura 17 – Especificação dos pinos do *66 Channel LS20031 GPS 10Hz Receiver*.  
(Sparkfun, 2006).

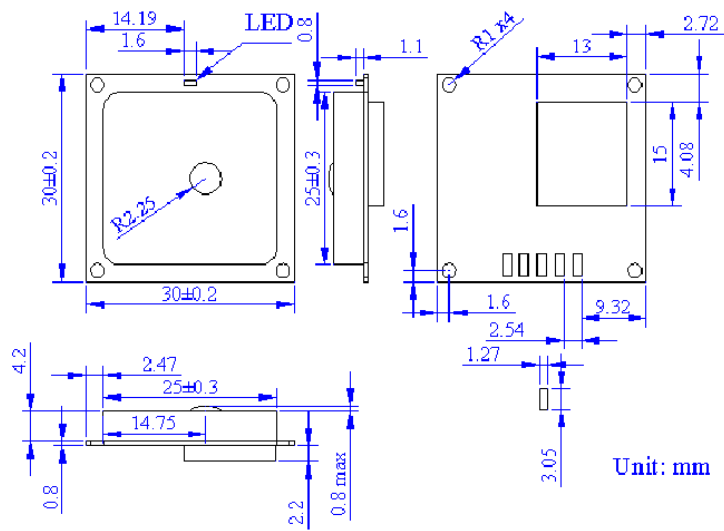


Figura 18 – Dimensões do 66 Channel LS20031 GPS 10Hz Receiver. (Sparkfun, 2006).

### 2.3.2. SISTEMA DE COMANDO E CONTROLO

O sistema de comando e controlo para este projeto tem como funcionalidade receber os dados do recetor de GPS, trata-los e enviar sinais para os motores. Para, de acordo com os dados, atuar e manter-se na posição ordenada.

### ARDUINO

O Arduíno é uma ferramenta de desenvolvimento *open source*<sup>6</sup>, tendo surgido de um projeto académico. O Arduíno está associado à criação de sistemas físicos baseados no uso de *software*<sup>7</sup> e *hardware* flexível e fácil de utilizar. (Santos, 2010).

As principais vantagens do Arduíno:

- Baixo custo;
- Simplicidade de utilização;
- Poder ser utilizado em vários ambientes (Windows, Macintosh e Linux).

O Arduíno permite ser utilizado como uma importante ferramenta de desenvolvimento, onde se pode fazer desde robots a domótica<sup>8</sup>, entre muitas outras aplicações. Bastando simplesmente ter imaginação, sendo a sua aprendizagem (com a correta dedicação) muito simples. (Arduino, 2006).

---

<sup>6</sup> O conceito de *open source* está associado a uma filosofia de partilha global em que todo o conteúdo pode ser visualizado e alterado sem qualquer restrição.

<sup>7</sup> *Software* é uma sequência de instruções escritas para serem interpretadas por um computador com o objetivo de executar tarefas específicas.

<sup>8</sup> A domótica é uma tecnologia que permite a gestão de todos os recursos habitacionais.

## ARDUINO MEGA 2560

Existem várias versões de Arduino, que utilizam diversos microcontroladores. A relação entre versões existentes e o microcontrolador está apresentada na Tabela 4 – Relação entre modelo de Arduino e o microcontrolador utilizado. (Santos, 2010).

<b>Modelo</b>	<b>Microcontrolador utilizado</b>
Arduino Duemilinueve	ATmega168 ou ATmega328
Arduino Diecimila	ATmega168
Arduino Mega	ATmega1280
Arduino Nano	ATmega168 ou ATmega328
LilyPad	ATmega168V
Pro	ATmega168 ou ATmega328
Pro mini	ATmega168

Tabela 4 – Relação entre modelo de Arduino<sup>9</sup> e o microcontrolador utilizado.

Analisando a tabela anterior, constata-se que existem quatro microcontroladores diferentes, utilizados pelos diversos modelos. As principais diferenças entre os microcontroladores são em termos de quantidades de memória disponível, e o consumo energético nas mesmas condições de funcionamento.

Os Arduino têm diferentes características uns dos outros. As principais características do Arduino Mega são apresentadas na Tabela 5. (Arduino, 2006).

---

<sup>9</sup> Mais informação acerca de todos os modelos no seu site oficial <http://www.arduino.cc>.

ARDUINO	MEGA
Microcontrolador	ATmega2560
Tensão de operação	5V
Tensão de entrada para alimentação (intervalo limite)	6-20V
Pinos de <i>input/ output</i> digital	54
Pinos analógicos	16
Pinos PWM <sup>10</sup>	15
Corrente DC por pino de <i>input/output</i>	40 mA
Corrente DC	50 mA

Tabela 5 – Características do Arduíno Mega. (Arduino, 2006).

Analisando a tabela anterior, podemos verificar que o Arduíno Mega apresenta um vasto conjunto de portos de entrada e de saída de sinal, sendo estes portos tanto analógicos como digitais. Esta característica permite a integração de uma maior diversidade de sensores nesta placa de desenvolvimento que os outros modelos apresentados na Tabela 4. Com a exceção da quantidade de pinos e do microcontrolador utilizado, as restantes características entre os diversos Arduíno não diferem uns dos outros.

No Arduíno Mega existe assim a possibilidade de ligar 16 sinais analógicos em simultâneo para efetuar a conversão A/D e, caso seja necessário, efetuar algum processamento e armazenamento.

Na Escola Naval, o sistema de comando e controlo mais utilizado é o Arduíno, devido também á sua facilidade de controlo e programação. Assim, para este trabalho vai ser usado controlador Arduíno, modelo Mega 2560.

<sup>10</sup> Iniciais utilizadas para designar *Pulse Width Modulation*. Este método permite variar a tensão média de um sinal de saída variando o seu *duty cycle* (tempo em que o sinal está no estado lógico 1).

A Figura 19 apresenta uma imagem frontal do mesmo Arduino em que se pode visualizar os diversos pinos e equipamentos associados.

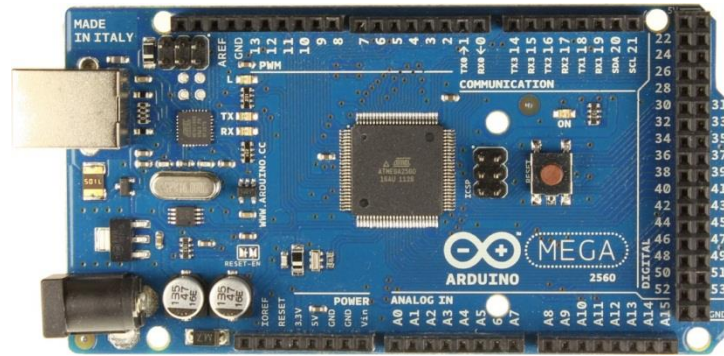


Figura 19 – Arduino Mega 2560. (Arduino, 2006).

### Pinout Arduino Mega 2560

Para o seguinte Arduino existem sete tipos de *pinout* disponíveis:

1) Pinos de I/O digitais: Sendo que alguns tem a capacidade de criar sinais PWM. A capacidade de utilizar PWM, é muito importante pois permite a partir de um sinal que só pode assumir o estado lógico 0 (0V) ou 1 (5V) obter um sinal analógico. O conceito de PWM é referido quando um sinal tem uma frequência constante mas um *duty cycle* variável;

2) Pinos analógicos: Para conseguir perceber os valores que obtidos tem que se analisar a conversão A/D feita. O microcontrolador utilizado possui um conversor analógico de 10 bits, ou seja:

$$2^{10}=1024$$

Como a tensão máxima de referência é 5 volts então:

$$1023 \leftrightarrow 5 \text{ volts}$$

$$0 \leftrightarrow 0 \text{ volts}$$

Logo,

$$1 \leftrightarrow x = \frac{5}{1023} \cong 0,00488 \cong 5mV$$

Com isto consegue-se concluir que o valor lido pelo Arduino só se vai alterar a cada 5mV de variação do sinal de entrada.

3) Pinos de *Ground*;

4) Pinos de saída para alimentar a 5 V;

5) Pinos de saída para alimentar a 3.3 V;

6) Pino de *Analog Reference* este pino permite mudar a referência analógica de 5V para o valor de entrada. Ficando todas as entradas analógicas com a referência introduzida;

7) Pinos de *reset* que servem para fazer *reset* ao Arduino;

8) Pino de  $V_{in}$  pode servir para alimentar o Arduino na gama de valores especificada, ou caso se alimente o Arduino, recorrendo a entrada USB ou a entrada tipo Jack, sendo que a alimentação do Jack recomendada e entre os 7 e os 12V. (Arduino, 2006).

# CAPÍTULO 3 - ARQUITETURA DO SISTEMA

---

Neste capítulo será descrito o trabalho desenvolvido. Este trabalho divide-se em três subcapítulos, *Hardware*, *Software* e *Interface*. No subcapítulo de *Hardware* são descritos todos os componentes utilizados para o desenvolvimento do projeto. No subcapítulo de *Software* é descrito apresentado toda a arquitetura do sistema, bem como o funcionamento do sistema de controlo automático. No subcapítulo *Interface* é descrito a ligação entre o *Hardware* e o *Software*.

## 3.1. HARDWARE

O sistema de controlo do projeto é composto por vários componentes que estão ligados ao Arduíno. Na Tabela 6 é apresentado a lista de componentes referentes ao Arduíno.

Protoshield PCB de Arduíno MEGA
Módulo Wi-Fi RN-XV WiFly
Módulo SD
Regulador de tensão 3.3V
Condensadores 1000uF
GPS LS20031

Tabela 6 – Componentes utilizados na placa Arduíno.

Inicialmente utilizou-se uma *breadboard*<sup>11</sup> para efetuar as ligações do *Recetor GPS LS20031* com o *Arduíno Mega*. Como o *Recetor GPS* necessita de uma tensão de entrada de 3.3V, utiliza-se um regulador de tensão para regular os 5V fornecidos pelo *Arduíno* para 3.3V. O Arduíno possui um pino de alimentação de 3.3V, no entanto utiliza-se o regulador de tensão e os condensadores para estabilizar a tensão fornecida para o *Recetor GPS*.

---

<sup>11</sup> A *breadboard* é uma placa de ensaios, ou seja, é utilizada para montar circuitos e testá-los.

O Módulo SD, que contém o cartão de memória, é ligado diretamente ao Arduino. Na Figura 20 está presente o Arduino com as ligações efetuadas para o Recetor GPS, o Módulo SD e o cartão de memória utilizado.

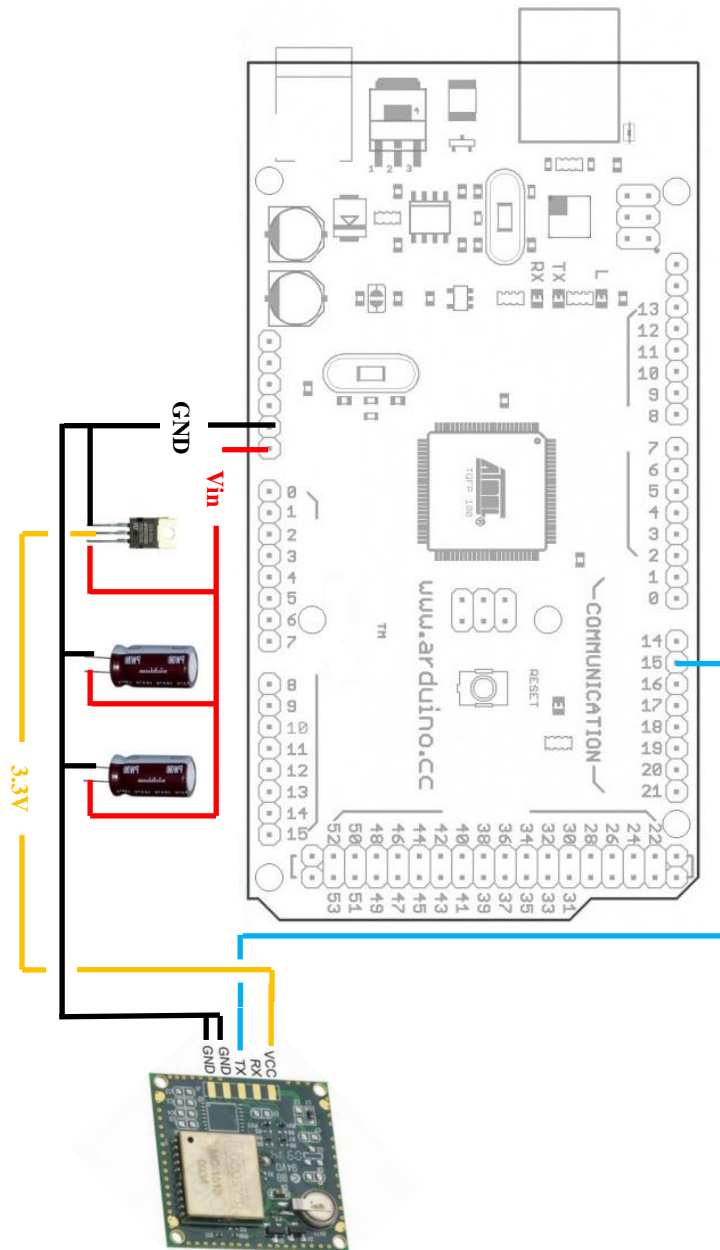


Figura 20 – Ligações dos componentes utilizados no Arduino.

Para testar o algoritmo de controlo houve a necessidade utilizar um protótipo. Deste modo a deslocação para realizar testes ao algoritmo de controlo torna-se mais fácil. A figura abaixo apresenta o protótipo utilizado.



Figura 21 – Protótipo utilizado para testar sistema de controlo.

A escolha deste protótipo baseou-se no projeto em tamanho real. O projeto em tamanho real é em forma de um catamaran e possui pouco calado. O sistema de direção também é o indicado, pois não possui leme. Este orienta-se segundo os seus dois motores posicionados a ré do projeto.

### 3.2. SOFTWARE

Esta secção tem como objetivo explicar o funcionamento do sistema de controlo automático. É apresentada a arquitetura do sistema, e posteriormente é especificado cada um dos blocos. Está também presente em Apêndice A e B a biblioteca e rotinas do Arduino. Em Apêndice D está presente o programa em suporte digital.

A figura abaixo indica a arquitetura geral do sistema de controlo.

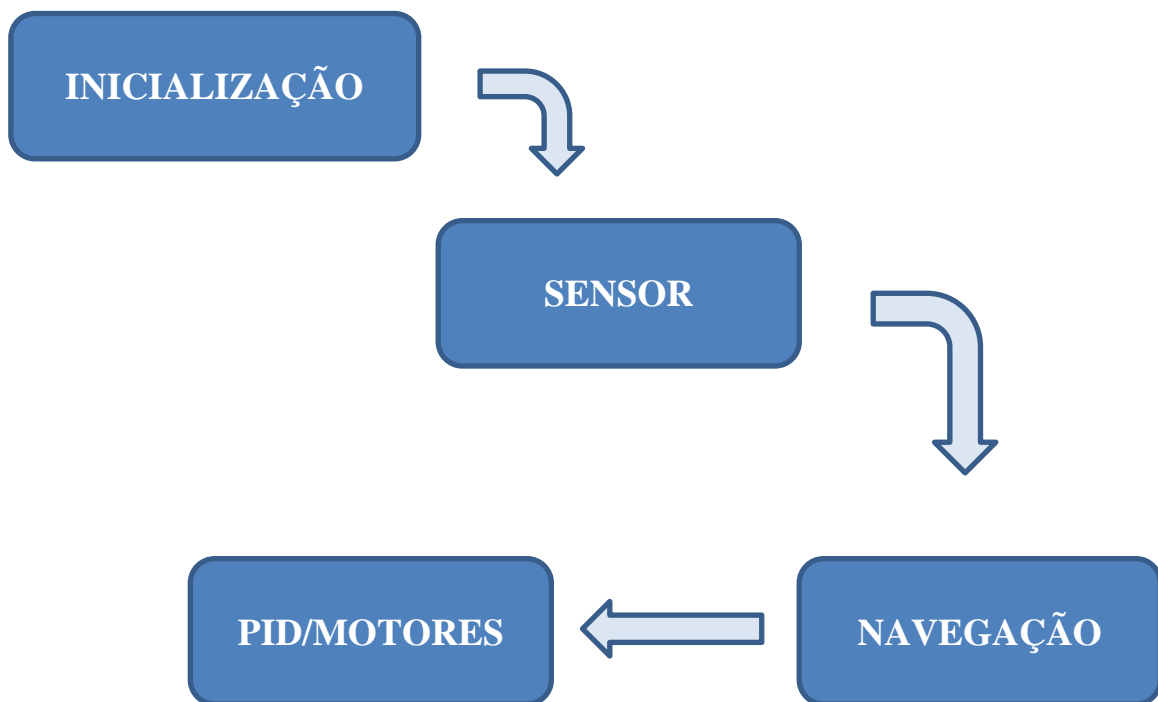


Figura 22 – Fluxograma geral do sistema de controlo.

#### INICIALIZAÇÃO

O bloco da inicialização tem um papel importante na fase inicial do programa. Este bloco vai permitir correr o resto do programa, sendo que para tal tem que declarar todas as variáveis necessárias para cada bloco acima referido na Figura 22, inicializar as comunicações série dos diversos sensores e, não menos importante, criar um ficheiro onde é guardada toda a informação previamente definida pelo programador. (Marques, 2012).

A Figura 23 representa o fluxograma deste bloco.

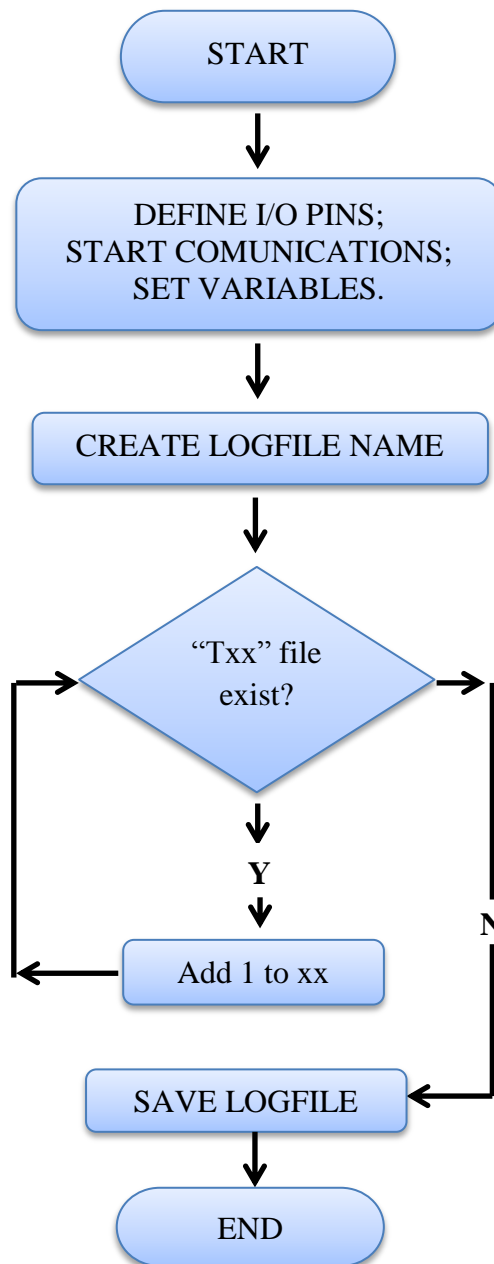


Figura 23 – Fluxograma do bloco de inicialização.

As rotinas que compõem o bloco de inicializações são as seguintes:

- Inicializar comunicações série entre:

- Arduíno com *Personal Computer* (PC) – INIT\_PC();

- GPS com Arduino – INIT\_GPS().

- Definir pinos para leitura/escrita de sinal do cartão SD – INIT\_SD();

- Definir o valor inicial das variáveis – INIT\_VAR();

- Definir nome do ficheiro de gravação de dados – VALID\_LOG().

## SENSOR

Este bloco corresponde à leitura do sensor de GPS. Este bloco é responsável por fazer a interligação entre o sensor GPS e o Arduino. Só assim se torna possível obter dados para processar e obter o posicionamento do projeto. A Figura 24 representa o fluxograma da leitura do GPS.

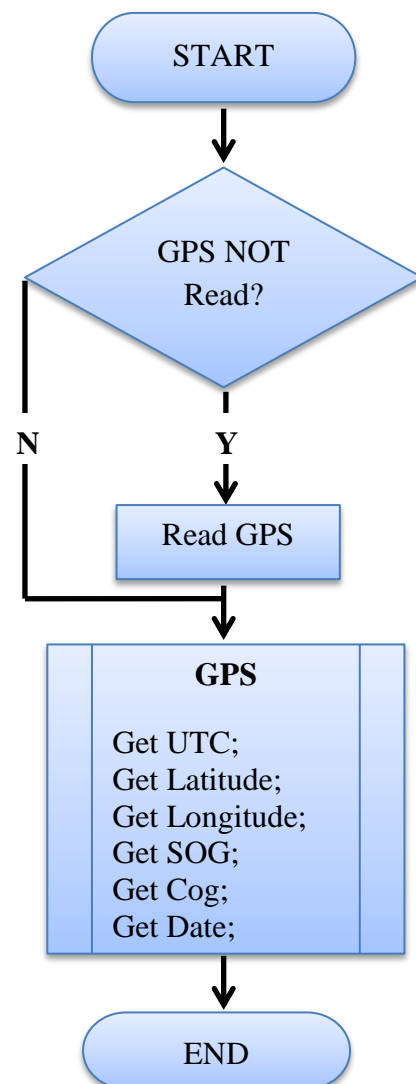


Figura 24 – Fluxograma da leitura de GPS.

Este fluxograma considera as seguintes funções:

- Ler e gravar em variáveis a frase do GPS – GET\_GPS();
- Processar os dados gravados e extrair informação do GPS:
  - *Universal Time Coordinated* (UTC) – GET\_UTC();
  - Latitude – GET\_LAT();
  - Longitude – GET\_LONG();
  - *Speed Over the Ground* (SOG) – GET\_SOG();
  - *Course Over the Ground* (COG) – GET\_COG();
  - Data – GET\_DATA().

## CALCULOS DE NAVEGAÇÃO

É no bloco de navegação que o programa com os dados das referências de posição, vai calcular a distância à posição desejada, e o seu azimute. Consoante a distância à posição desejada este bloco vai atuar (ligar ou desligar) os motores. A Figura 25 representa o fluxograma do bloco de navegação.

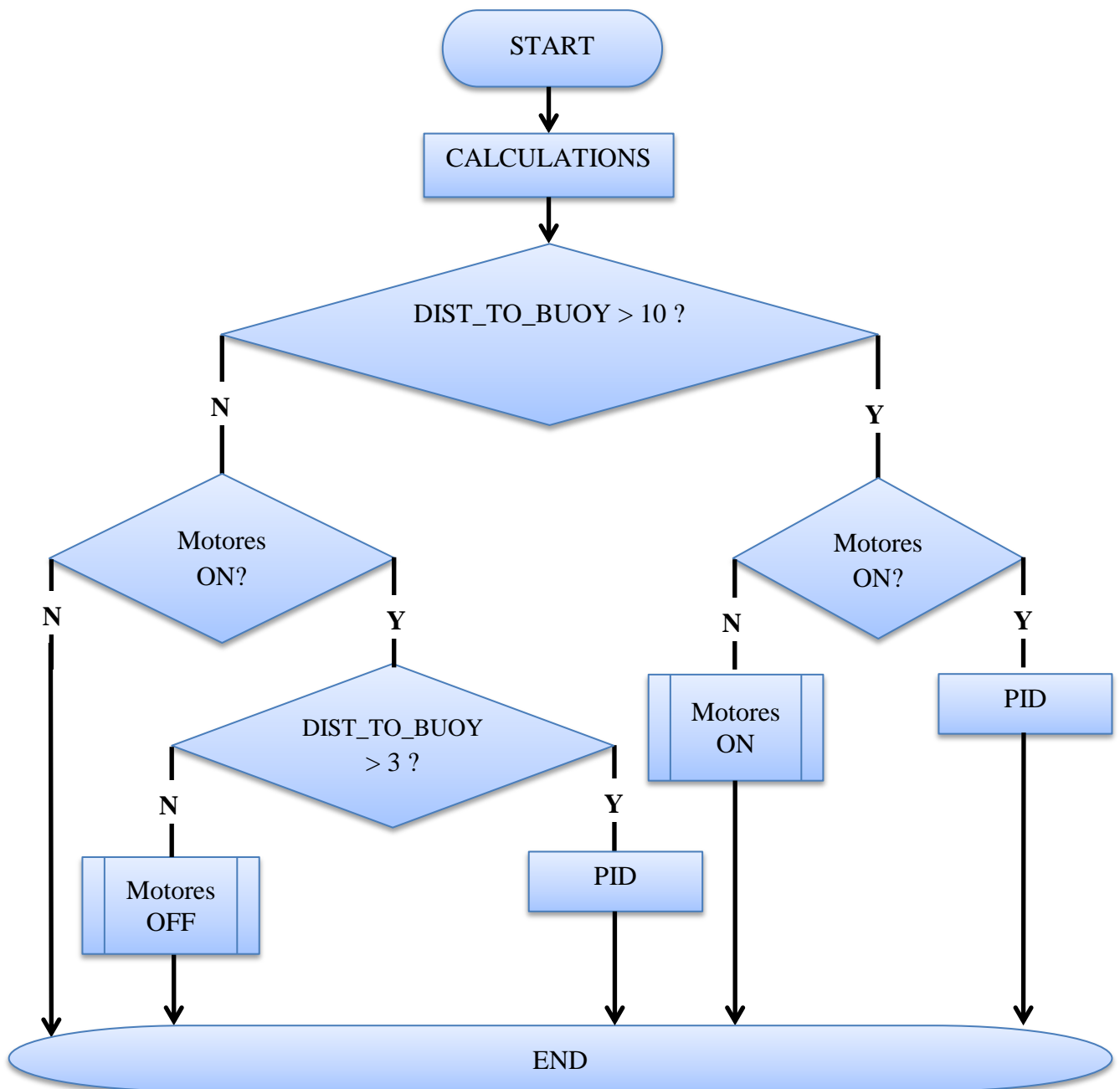


Figura 25 – Fluxograma do bloco de navegação.

Este fluxograma considera as seguintes funções:

- Efetuar cálculos de navegação no bloco CALCULATIONS:

- Distância à posição – GET\_BUOY\_DIST();

- Azimute à posição – GET\_BTWP().

- Desligar e ligar os motores consoante a distância à posição pretendida.

## CONTROLO PID/MOTORES

O bloco do controlo PID/MOTORES tem como objetivo variar os valores de PWM dos motores. Estes valores são calculados consoante um erro (diferença entre a proa e o azimute desejado). Deste modo, a boia irá manter-se sempre no rumo desejado como ilustra a Figura 26.

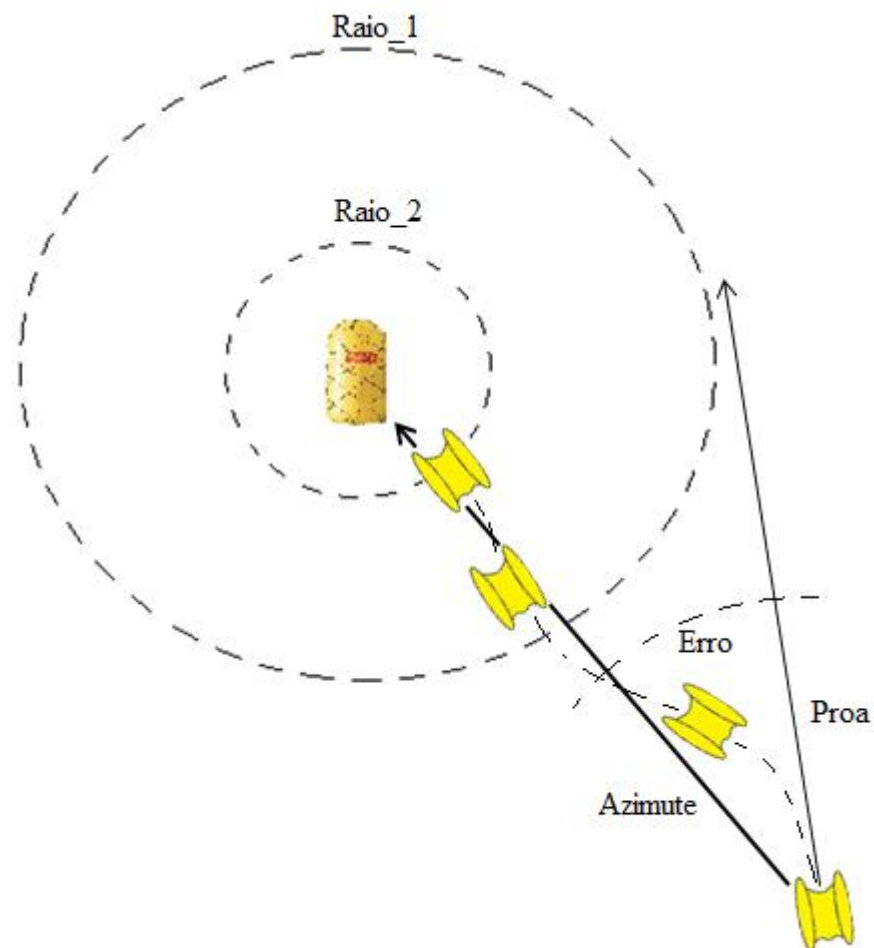


Figura 26 – Efeito do controlo PID.

Através do controlador PID, o erro vai diminuindo até se encontrar uma relação entre os dois motores que mantenha o projeto no azimute correto. Os motores são ligados através do bloco de Navegação, quando o projeto estiver a uma distância superior ao *Raio\_1*, e quando estiver a uma distância inferior ao *Raio\_2*, desliga os motores.

A Figura 27 representa o fluxograma do bloco PID/MOTORES.

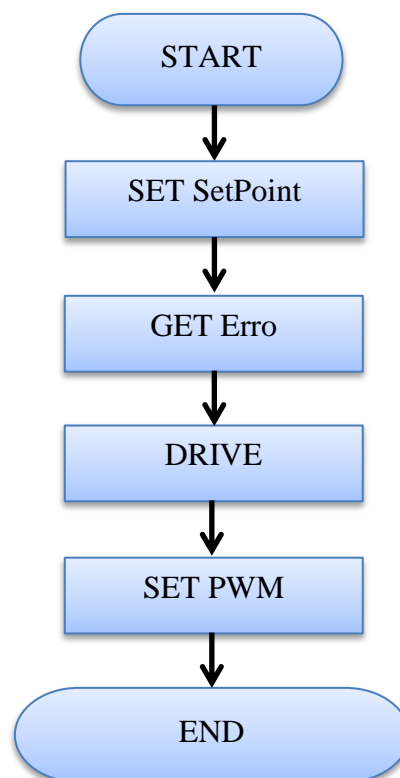


Figura 27 – Fluxograma do bloco PID/MOTORES.

Este fluxograma considera as seguintes funções:

- Calcular o erro – GET\_Marcacao();
- Calcular o diferencial dos motores – GET\_DRIVE();
- Atuar o PWM calculado nos motores.

### 3.3. INTERFACE

Neste sistema de posicionamento dinâmico é necessário fazer a ligação entre o sistema de controlo e o sistema de propulsão - *INTERFACE*. É neste subcapítulo que é apresentada a ligação entre estes sistemas.

É através do Arduino Mega que vão sair os sinais para o sistema de propulsão. Estes sinais são adaptados consoante as indicações de velocidade que o sistema de comando calcula. A função correspondente à saída destes sinais por parte do Arduino é a função *analogWrite()*.

Esta função possibilita a utilização dos pinos PWM da placa Arduino. O sinal PWM mantém-se até ser modificado através de outra instrução que afete esse mesmo pino. A frequência do sinal PWM criado é de  $\approx 490$  Hz. (Pessanha, 2006).

SINTAXE:

`analogWrite(número do pino, valor);`

O “número do pino” serve para indicar qual o pino que a função vai afetar. O “valor” varia entre 0 (sempre desligado), até 255 (representa um sinal de 5V constantes). (Pessanha, 2006).

Para variar a velocidade dos motores é necessário alterar o “valor”, alterando assim a largura de pulso e conseqüentemente o *Duty Cycle*<sup>12</sup>. A Figura 28 apresenta vários *Duty Cycle's* consoante o “valor” inserido na função *analogWrite()*. As linhas verdes representam um período de tempo regular de 2 milissegundos cada. Esta duração ou período é inversa à frequência do PWM. (Arduino, 2006).

---

<sup>12</sup> Duty Cycle é a proporção de tempo durante o qual um componente, dispositivo ou sistema está em operação, neste caso em 5V.

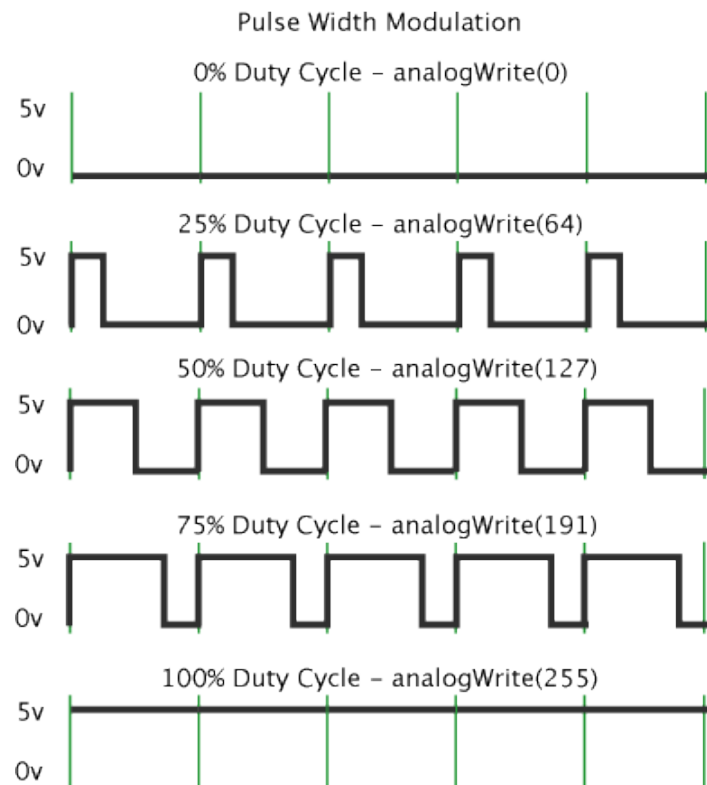


Figura 28 – *Pulse Width Modulation* – PWM. (Arduino, 2006).

Para este sistema de posicionamento dinâmico é necessário várias velocidades para os motores tanto no sentido positivo como no negativo. Com o controlo PID, o sistema vai variando o *Duty Cycle* à medida que é necessário para se posicionar.

Para inverter o hélice do motor, inverte-se o sinal de 5V como apresenta a Figura 29 para um *Duty Cycle* de 25%.

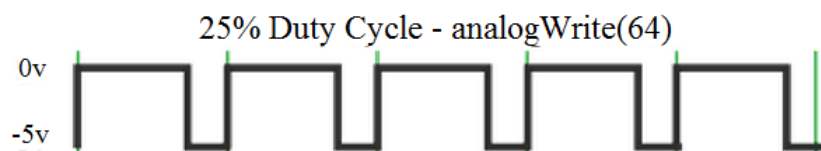


Figura 29 – *Duty Cycle* 25%, velocidade no sentido negativo. (Arduino, 2006).

A figura seguinte ilustra o controlador de motores de corrente contínua projetado e construído pelo Aspirante EN-AEL Jorge de Jesus.

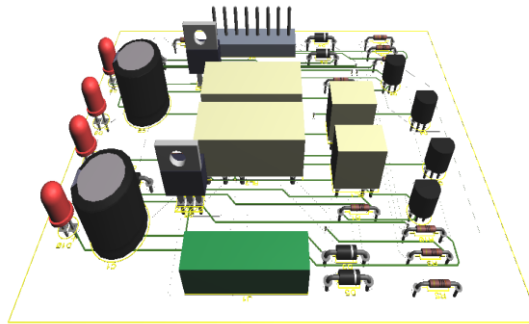


Figura 30 – Controlador de motores de corrente contínua.

Os sinais PWM provenientes do Arduino Mega quando são encaminhados para o sistema de propulsão passam por um controlador de motores de corrente contínua.

A figura seguinte representa a malha do sistema de posicionamento dinâmico baseado em GPS.



Figura 31 – Malha do sistema de posicionamento dinâmico baseado em GPS.

A figura anterior ilustra todo o sistema de posicionamento dinâmico baseado em GPS. Este compreende um sistema de referência que transfere para o sistema de controlo os dados relativos à posição geográfica (latitude e longitude). O sistema de controlo após efetuar os cálculos necessários emite sinais PWM para o sistema de propulsão constituído por dois motores elétricos (BB e EB).

# CAPÍTULO 4 – SIMULADOR E TESTES

---

Neste capítulo será abordada toda a estrutura do simulador, assim como a implementação dos dois algoritmos descritos anteriormente e o cenário de teste. Posteriormente serão descritos os vários testes realizados, com o protótipo, e suas conclusões.

## 4.1. SIMULADOR

Para o desenvolvimento deste projeto houve necessidade de criar um simulador com a finalidade de testar os algoritmos de posicionamento. Assim, ao criar um simulador pode-se ajustar/modificar algoritmos, facilitando depois a criação de um algoritmo final para introduzir no sistema de comando e controle deste projeto.

Para tal, foi escolhido o *Matlab*<sup>13</sup> *R2011b*. O *Matlab* é uma linguagem de programação apropriada ao desenvolvimento de aplicativos de natureza técnica. É adequado para implementar e testar soluções com facilidade e precisão. Para isso, possui facilidades de computação, visualização e programação, de fácil aprendizagem. A figura seguinte representa a interface gráfica do simulador em *Matlab*.

A programação relativa a este simulador pode ser visualizada no CD-ROM que está presente no Apêndice D.

---

<sup>13</sup> Matlab - é um *software* interativo de alta performance voltado para o cálculo numérico.

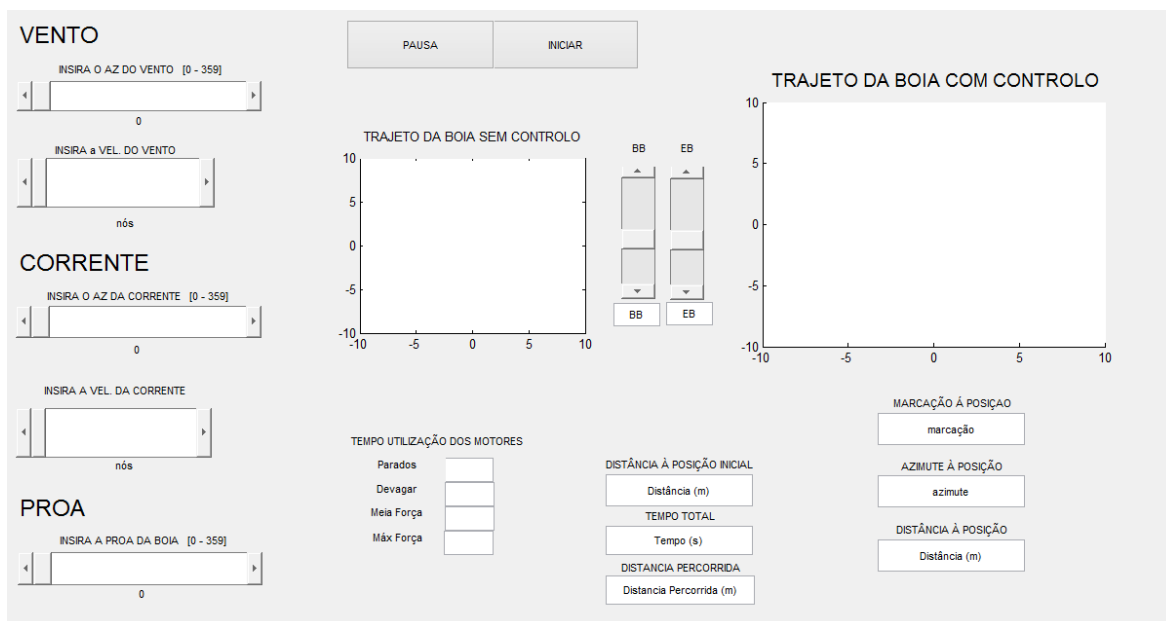


Figura 32 – Interface gráfica do simulador em *Matlab (Versão R2011b)*.

Como se pode visualizar nesta figura, esta interface está dividida em três campos. O primeiro é relativo a alterações de variáveis (vento, corrente e proa). Estas variáveis podem ser alteradas pelo utilizador a qualquer momento.

O segundo campo é constituído por dois gráficos. Um gráfico representa a trajetória da boia com o sistema de controlo. O outro gráfico representa a trajetória da boia sem qualquer tipo de controlo, simplesmente é afetada pelas perturbações (vento e corrente).

O terceiro campo é constituído pela apresentação de várias variáveis que estão em constante mudança. Estas variáveis (estado dos motores, tempo de utilização dos motores, distância à posição, azimute à posição...) permitem ao utilizador visualizar o que está a acontecer com a boia.

Na Figura 33 é apresentado o diagrama de blocos do simulador desenvolvido e de seguida explicado sucintamente em que consiste cada um dos blocos.



Figura 33 – Diagrama de blocos do simulador.

#### INICIALIZAÇÃO:

No primeiro bloco da Figura 33 é onde são declaradas e inicializadas as variáveis (eixos, motores, posição inicial,...).

#### PERTURBAÇÕES:

Na interface gráfica existem variáveis que o utilizador pode introduzir ou alterar no decorrer do programa, relativamente às perturbações e proa da boia. É neste bloco que o simulador vai adquirir o valor destas variáveis introduzidas pelo utilizador.

#### POSICIONAMENTO:

Este bloco consiste em dois tipos de posicionamento. Posicionamento que só é afetado pelas condições atmosféricas para visualizar o que aconteceria à boia caso não tivesse

qualquer tipo de controlo. E posicionamento que engloba todas as condições, tanto atmosféricas como a variação da proa, e métodos de controlo de posição para não se afastar da sua posição inicial.

Com a atualização destas posições é calculado, a cada segundo, a distância da posição atual até à posição inicial, tanto para a posição sem qualquer tipo de controlo como para a com controlo.

Este bloco de posicionamento engloba outros algoritmos como o controlo dos motores e o cálculo da proa da boia.

#### GRÁFICOS:

Este bloco gráficos tem como funcionalidade reproduzir os posicionamentos calculados no bloco posicionamento na interface gráfica. Na interface gráfica vão aparecer dois gráficos, um com o posicionamento com o sistema de controlo e outro com o posicionamento simplesmente afetado pelas perturbações.

Por fim, existe uma função que pára todo o ciclo, associada ao botão “PARAR” representado na Figura 30. Embora esta função pare o ciclo, os gráficos só serão limpos caso se inicie outra vez o programa.

#### 4.1.1. IMPLEMENTAÇÃO DOS ALGORITMOS DE POSICIONAMENTO

A boia quando posicionada no mar está sujeita a perturbações induzidas pelas condições atmosféricas (correntes, ventos...) afetando o seu posicionamento e consequentemente aumentando o erro de posição.

Para testar os algoritmos de posicionamento, estes implementaram-se no simulador. Assim é possível observar o seu comportamento e tirar conclusões. Se necessário adequá-los às necessidades.

## Cenário de teste

Para a realização de testes dos algoritmos é necessário criar um cenário igual para o algoritmo *way-point* e *line-tracking*. Assim é possível comparar e obter uma conclusão. Para tal, de modo a obter o máximo de conclusões, faz-se vários testes alterando a orientação das perturbações. Deste modo, consegue-se analisar o comportamento da boia e alterações de trajetória para cada *orientação das perturbações*.

A Figura 34 representa os vários cenários a que os algoritmos se testaram.

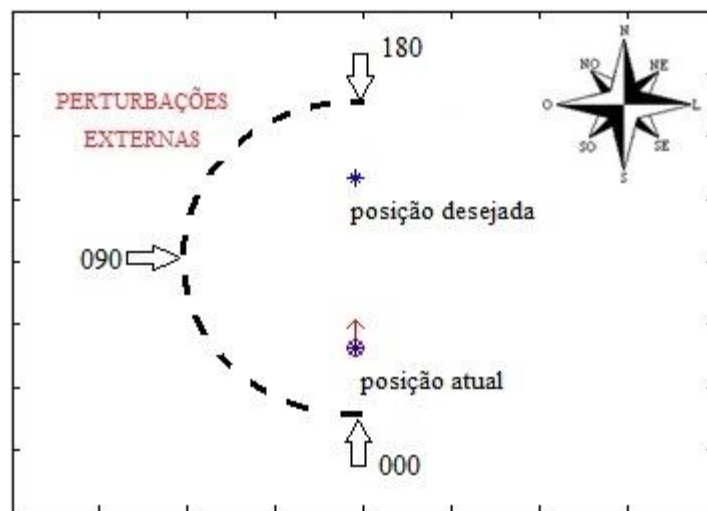


Figura 34 – Cenário das simulações.

Como se pode observar na Figura 34, a boia inicia a sua trajetória na *posição atual* e termina na *posição desejada*, uma distância de 20 metros.

O que vai afetar a trajetória da boia são as *perturbações externas* que vão variar a sua orientação de 000 a 180 em intervalos de 10 graus e com uma velocidade constante de 2 nós.

#### 4.1.2. SIMULAÇÃO DO ALGORITMO *WAY-POINT*

O algoritmo *way-point* simplesmente calcula o ângulo do ponto atual em relação ao ponto desejado. Depois é necessário saber em que quadrante se encontra para calcular o azimute. Para isso, utiliza-se o sinal do numerador ( $x_d - x$ ) e do denominador ( $y_d - y$ ) da função.

Na Tabela 7 observar-se o comportamento da boia com o algoritmo *way-point* para as diferentes orientações de forças externas. Esta tabela indica o tempo que a boia demora a deslocar-se para a posição destinada, a distância que percorre até lá chegar e a distância entre a posição final e a posição destinada. Também indica a percentagem de velocidades que utiliza para alcançar a posição final.

### ALGORITMO WAY-POINT

Orientação da perturbação	Tempo (s)	Distância percorrida (m)	Distância ao $W_p$ (m)	Parados	Devagar	Meia força	Máxima força
000	8	21,20	1,20	13%	88%	0%	0%
010	7	19,08	1,13	14%	79%	7%	0%
020	7	19,80	1,18	14%	71%	14%	0%
030	7	21,23	1,29	14%	64%	14%	7%
040	7	20,69	1,70	14%	64%	14%	7%
050	7	21,56	1,18	14%	57%	14%	14%
060	7	20,74	1,05	14%	57%	14%	14%
070	7	19,84	1,66	14%	57%	14%	14%
080	8	21,43	0,82	19%	50%	13%	19%
090	9	22,42	2,27	11%	61%	17%	11%
100	9	21,04	1,54	11%	61%	17%	11%
110	11	23,30	1,84	14%	59%	14%	14%
120	11	22,16	0,84	9%	64%	18%	9%
130	12	21,17	0,68	8%	67%	21%	4%
140	13	21,11	0,07	8%	69%	19%	4%
150	15	20,19	0,69	7%	77%	17%	0%
160	17	20,35	0,15	6%	82%	12%	0%
170	19	19,72	0,46	5%	89%	5%	0%
180	21	19,42	0,58	5%	95%	0%	0%
Média	10,63	20,87	1,07	11%	69%	13%	7%

Tabela 7 – Simulação do algoritmo *way-point*.

Após realizar a simulação do algoritmo *way-point* para todos os cenários da Tabela 7, é espetável que o *tempo* vá aumentando consoante a *orientação da perturbação* se aproxima de 180 (perturbação contrária à deslocação da boia). A única situação em que o *tempo* não aumenta com o aumento da *orientação da perturbação* é na passagem da orientação 000

para a 010. Isto deve-se ao facto de com a orientação de 000, a boia desloca-se com uma velocidade maior (devido à perturbação ter o mesmo sentido da deslocação da boia), e assim passa a posição pretendida para obter uma menor *distância ao Wp*.

Na coluna da *distância percorrida* não há grandes alterações, varia aproximadamente entre 19 e 21 metros, exceto quando a *orientação da perturbação* é perpendicular ou se aproxima da perpendicular à deslocação da boia, que se visualiza distâncias maiores (valores entre os 21 e os 23 metros, ocorre para as *orientações de perturbação* de 080 a 120), e faz com que a boia se afaste mais da sua trajetória ideal.

A *distância ao Wp* é a distância mais próxima que a boia se aproxima da posição pretendida. Para estes testes efetuados obteve-se na melhor situação 0,07 metros e para a pior 2,27 metros de *distância ao Wp*.

As últimas colunas da Tabela 7 são referentes à utilização dos motores. A maior utilização dos motores é em modo *devagar* com uma média de 69%. A boia utiliza o modo *devagar* para se deslocar, e os outros modos para guinar com os motores e alterar a sua proa. Assim, pode-se observar que pela percentagem de utilização dos outros modos dos motores, a boia está em constante mudança de proa como é espetável. Na coluna de *máxima força*, observa-se uma maior percentagem de utilização para os valores de *orientação da perturbação* de 050 a 120 (valores que se aproximam à perpendicular da trajetória da boia). Nestes casos, a boia tem necessidade de alterar bastante a proa, e para tal utiliza a *máxima força*.

#### 4.1.3. SIMULAÇÃO DO ALGORITMO *LINE-TRACKING*

O algoritmo *line-tracking* pretende anular as forças externas. Para isso calcula o ângulo entre o ponto seguinte em que se deveria encontrar a boia, e onde realmente se encontra. Depois, idêntico ao algoritmo *way-point*, é necessário saber em que quadrante se encontra

para calcular o azimute. Na Tabela 8 pode-se observar o comportamento da boia para a simulação do algoritmo *line-tracking*.

### ALGORITMO *LINE-TRACKING*

Orientação da perturbação	Tempo (s)	Distância percorrida (m)	Distância ao $W_p$ (m)	Parados	Devagar	Meia força	Máxima força
000	8	21,20	1,20	13%	88%	0%	0%
010	8	21,13	1,66	13%	88%	0%	0%
020	7	19,68	0,45	14%	71%	14%	0%
030	7	19,27	1,50	14%	71%	14%	0%
040	7	19,54	1,49	14%	64%	21%	0%
050	7	18,82	2,33	14%	71%	7%	7%
060	7	19,48	1,05	14%	64%	7%	14%
070	8	20,72	1,30	13%	69%	6%	13%
080	8	19,56	1,78	13%	69%	6%	13%
090	10	20,05	1,47	15%	70%	5%	10%
100	11	20,47	1,34	14%	73%	5%	9%
110	12	20,12	0,89	13%	75%	4%	8%
120	13	19,59	0,87	12%	77%	4%	8%
130	15	20,13	1,83	10%	80%	3%	7%
140	16	19,74	4,09	9%	81%	3%	6%
150	15	19,12	4,63	7%	83%	3%	7%
160	14	19,87	4,27	7%	75%	11%	7%
170	18	18,54	3,12	6%	89%	6%	0%
180	21	19,42	0,58	5%	95%	0%	0%
MÉDIA	11,16	19,81	1,89	12%	76%	6%	6%

Tabela 8 - Simulação do algoritmo *line-tracking*.

Analisando a Tabela 8, simulação do algoritmo *line-tracking* para os vários cenários, observa-se e como exetável que o *tempo* vai aumentando consoante aumenta também a

*orientação da perturbação*, exceto para as orientações 000, 010, 150 e 160. Nas situações 000 e 010, à semelhança do algoritmo *way-point*, a boia passa a posição pretendida para obter uma menor *distância ao Wp*. Nas situações 150 e 160, não ocorre o que é previsto porque a boia posiciona-se ligeiramente antes do *Wp*.

Os valores da *distância percorrida*, como se pode esperar, não diferem muito. O pouco que variam é devido à distância que a boia utiliza para alterar a proa inicialmente com a variação dos motores.

A *distância ao Wp* vai sempre variar porque a boia só altera a sua proa inicialmente, tendo em conta com as perturbações. Como a proa varia com os motores, a boia tem dificuldade em colocar a proa num determinado valor. Assim, só que a proa fique com um erro de um ou dois graus, quando chegar á sua posição final tem uma distância ao *Wp* significativa. É o que se verifica nas perturbações 140, 150, 160 e 170.

As últimas colunas da Tabela 8 são referentes à utilização dos motores. Verifica-se como é de esperar que grande percentagem dos motores é em modo *devagar*. A boia só vai inicialmente utilizar os outros modos para guinar. Todo o resto do percurso até à sua posição final é em modo *devagar*. Também é de esperar que a boia utilize uma maior percentagem de *máxima força* quando a *orientação da perturbação* se aproxima da perpendicular à trajetória da boia, o que se verifica para os valores 060, 070, 080, 090 e 100 da *orientação da perturbação*.

#### 4.1.4. CONCLUSÕES

Para o algoritmo *way-point* a boia vai atualizando, a cada momento, a sua orientação sem ter em conta com as perturbações externas. Isto, no caso de existirem perturbações com uma orientação diferente ou inversa à da boia, faz com que a boia se afaste do seu rumo ideal e faça uma trajetória curvilínea até à sua posição final.

No caso do algoritmo *line-tracking*, a boia afasta-se caso haja perturbações externas, e logo no instante seguinte altera a sua orientação tendo em conta essas perturbações, seguindo um rumo constante.

Ao comparar-se os resultados obtidos das diversas simulações, Tabela 7 e Tabela 8, observa-se que ao utilizar o algoritmo *way-point* a distância percorrida pela boia vai ser maior devido à sua curvatura até atingir a posição desejada. Este algoritmo também tem como desvantagem o facto de ter uma maior percentagem na utilização de maiores velocidades, o que vai implicar maiores consumos de energia. Como o algoritmo *line-tracking* só necessita de alterar o estado dos motores inicialmente para alterar a sua proa, vai ter um aproveitamento de energia melhor que o algoritmo *way-point* que tem que estar sempre a atualizar a proa e a alterar o estado dos motores.

A grande vantagem do algoritmo *way-point* é o facto de ter uma aproximação melhor que o algoritmo *line-tracking*. Isto porque o algoritmo *line-tracking* não tem em conta a distância que se afasta inicialmente, para poder calcular uma orientação de referência.

Depois das simulações realizadas pode-se afirmar que o algoritmo *line-tracking* é mais indicado para efetuar trajetos de longas distâncias e para situações em que as perturbações são constantes. No entanto, nas situações de constantes alterações das perturbações externas, o algoritmo não é muito eficaz. Enquanto, que o algoritmo *way-point* é mais eficaz para situações de mudanças de perturbações, e para curtas distâncias.

Neste caso, como é desejável que a boia se mantenha numa posição definida ou próximo dela, é preferível o algoritmo *way-point*, em que atualiza constantemente a posição atual independentemente das perturbações alterarem ou se manterem.

## 4.2. TESTES

De forma a confirmar o desempenho do sistema de controlo desenvolvido foram efetuados vários testes. Numa primeira fase foram efetuados testes aos equipamentos, em ambiente seco. Só mais tarde, foram efetuados testes ao código do sistema tanto em ambiente seco como na água.

Segue abaixo os testes efetuados, bem como os locais e datas:

- Teste ao *Recetor GPS LS20031*:
  - No Departamento de Armas e Eletrónica da Escola de Tecnologias Navais dia 9 de maio de 2013;
  - Na piscina da Base Naval de Lisboa (BNL) dia 11 de maio de 2013.
- Testes ao controlador PID:
  - Em águas paradas de dia 10 a 16 de agosto de 2013;
  - No rio com corrente de dia 17 a 23 de agosto de 2013.

Nos testes acima descritos, excluem-se os testes realizados em laboratório. Os testes realizados em laboratório têm como função testar as ligações feitas e o correto funcionamento do código entre sistemas. Só depois se puderam realizar os testes acima mencionados.

Após cada teste efetuado, os dados referentes ao Recetor GPS ficam gravados num ficheiro no cartão SD. Para visualização desses dados, importa-se o ficheiro para o Google Earth versão 7.1.1.1888 e é indicado o percurso efetuado pelo sistema.

#### 4.1.1 TESTES AO RECETOR GPS LS20031

De modo a iniciar a programação do controlador PID, realizaram-se testes ao recetor GPS para verificar o seu funcionamento, assim como a gravação de dados para o cartão de memória. Segue abaixo os testes realizados.

- ESCOLA DE TECNOLOGIAS NAVAIS DIA 09 DE MAIO DE 2013

Primeiramente, para verificar o bom funcionamento do *Recetor GPS LS20031*, foi realizado um teste num percurso à volta do Departamento de Armas e Eletrónica da Escola de Tecnologias Navais.

A Figura 35 mostra através do Google Earth, o percurso efetuado pelo sensor *gps*.



Figura 35 – Teste realizado no Departamento de Armas e Eletrónica na Escola de Tecnologias Navais. (Google Earth).

Este teste foi efetuado com o *gps* no interior de um carro, e de seguida foi dada uma volta ao edifício. No ponto de partida do percurso, observa-se uma deformação que se deve ao

facto de se ter iniciado o *gps* aquando o carro ainda estava estacionado no parque e por fim voltado a estacionar no mesmo sítio.

Assim, conclui-se que o *Recetor GPS LS20031* está a funcionar como pretendido, bem como o programa que recebe e grava os dados provenientes do sensor.

- PISCINA DA BASE NAVAL DE LISBOA DIA 11 DE MAIO DE 2013

Após se ter realizado um teste com o *Recetor GPS LS20031*, e se ter verificado o seu correto funcionamento, houve a necessidade de se realizar outro teste. Agora, num local de menores dimensões, para verificar o seu erro. Escolheu-se a piscina da Base Naval de Lisboa para verificar o erro não só por ser um local mais pequeno, mas também porque é o local onde mais tarde se irá realizar os primeiros testes do programa do Arduíno com o protótipo.

A Figura 36 mostra através do Google Earth o percurso efetuado pelo sensor *gps* em torno da piscina.



Figura 36 – Teste realizado na Piscina da Base Naval de Lisboa. (Google Earth).

Este teste realizou-se em torno da piscina da Base Naval de Lisboa, fora de água. Iniciou-se o teste na posição do Wp\_01, seguindo os outros por ordem numérica até regressar ao início.

Como se pode observar na figura acima, o *Recetor GPS LS20031* tem um erro em relação à imagem do *Google Earth*. No entanto, a forma da piscina mantêm-se a mesma, o que significa que as posições recebidas pelo *gps* são sequenciais e sem oscilações ou erros de posição.

#### 4.1.2 TESTES AO CONTROLADOR PID

Depois de realizar os testes acima referidos, e de programar o controlador PID, foi necessário realizar uma série de testes para verificar o funcionamento do controlador e proceder a reajustes do mesmo. Estes testes foram realizados em dois cenários diferentes, em águas paradas e no rio onde existem perturbações (corrente e vento). Para estes dois cenários, realizaram-se testes para curtas e longas distâncias.

À semelhança dos testes efetuados com o recetor GPS, estes também foram visualizados no *Google Earth* para visualizar o trajeto percorrido pelo protótipo. Nestas figuras, a posição desejada está assinalada com um marcador denominado *Wp*.

- EM AGUAS PARADAS, DE DIA 10 A 16 DE AGOSTO DE 2013

Estes testes foram realizados numa presa de água do rio Nabão, no Agroal, situado na localidade denominada de Formigais. Esta presa de água é um local com dimensões suficientes para testar o controlador PID no protótipo. Neste local efetuou-se diversos testes, entre os quais se destacam os abaixo assinalados.

A figura abaixo representa o primeiro teste efetuado.



Figura 37 – Primeiro teste ao controlador PID, realizado em águas paradas. (Google Earth).

Como se pode observar através da figura acima referida, o controlador PID possui muitos erros. O protótipo não é capaz de encontrar a posição desejada. É também possível observar nesta imagem um erro de GPS, onde a imagem marca uma reta a partir do  $W_p$  até uma das margens da presa do rio.

A figura abaixo representa outro teste efetuado após diversos testes e reajustes no controlador PID.



Figura 38 – Teste ao controlador PID, após reajustes na programação. (Google Earth).

Este teste encontra-se desfasado da imagem do Google Earth, como acontece na Figura 36. Analisando o trajeto percorrido pelo protótipo pode-se dizer que percorreu uma trajetória circular. Isto deve-se ao facto de as constantes do controlo PID ( $K_p$ ,  $K_i$  e  $K_d$ ) não estarem ajustadas de forma correta para este protótipo.

Desta forma procede-se ao acerto das constantes. A figura abaixo representa o teste efetuado após este acerto das constantes.

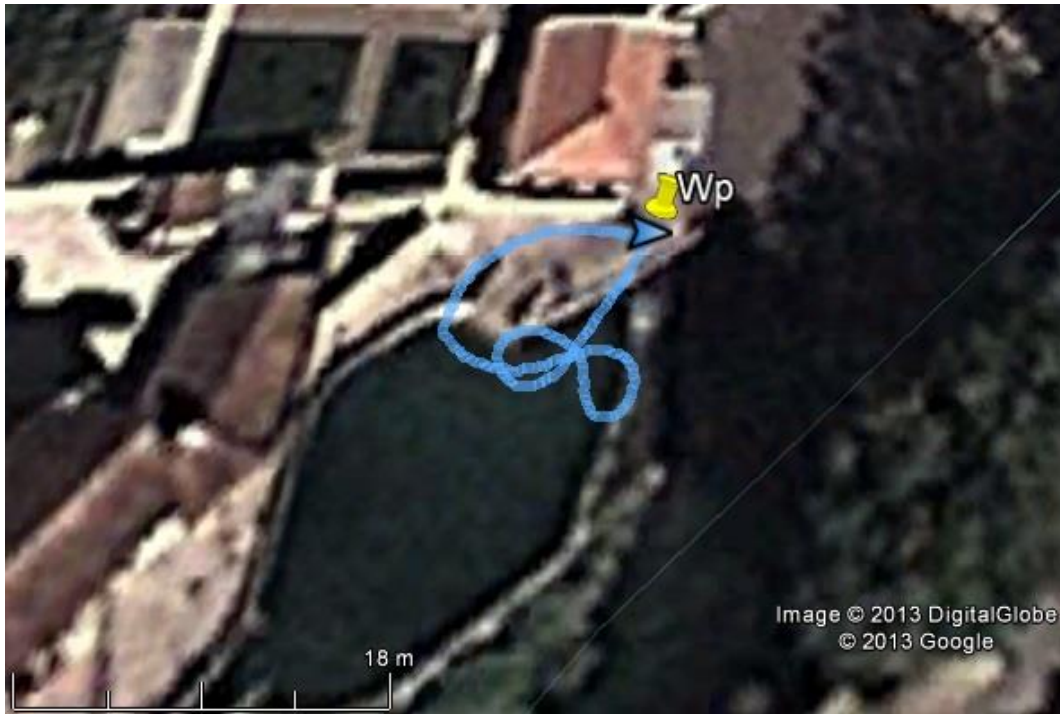


Figura 39 – Teste ao controlador PID, após reajustes nas contantes ( $K_p$ ,  $K_i$  e  $K_d$ ). (Google Earth).

Nesta figura, embora também desfasada da imagem do Google Earth, observa-se uma melhoria em relação às anteriores. Numa fase inicial (depois de se afastar do  $W_p$ ), o protótipo realiza uma trajetória em forma de oito, até conseguir encontrar o rumo correto.

Isto demonstra que as contantes do controlador PID, ainda não estão totalmente ajustadas.

A Figura 40 representa outro teste, depois de se realizar alguns testes para ajustar as contantes do controlo PID.

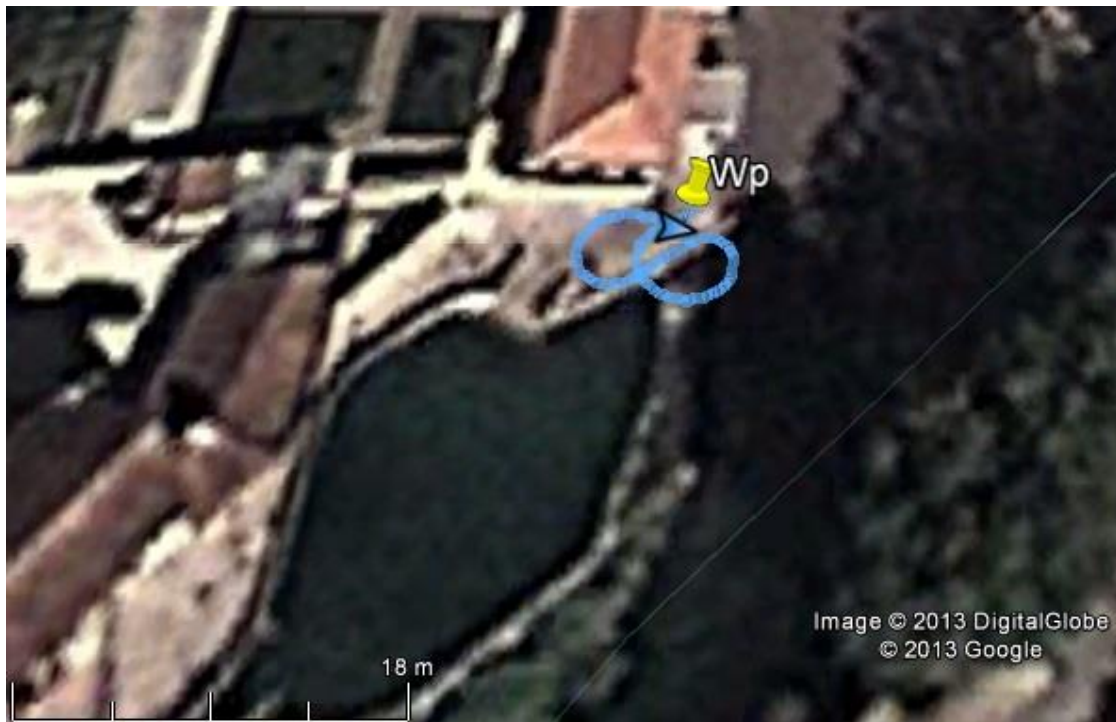


Figura 40 – Teste ao controlador PID, com melhoramento das contantes ( $K_p$ ,  $K_i$  e  $K_d$ ).  
(Google Earth).

Após muitos testes, pode-se afirmar que este foi concluído com sucesso. O protótipo afastou-se por duas vezes da posição desejada, e conseguiu nas duas vezes regressar à mesma posição.

É de salientar que estes testes realizados nem sempre são concluídos com sucesso devido também a erros de GPS, porque é uma zona de bastantes árvores. Por vezes, o GPS indica posições bastante longínquas de onde se encontra.

Outro problema destes testes é o facto da bateria que alimenta o protótipo ter uma curta duração (cerca de uma hora). Assim, impossibilita a realização de vários testes no mesmo dia.

- **NO RIO, DE DIA 17 A 23 DE AGOSTO DE 2013**

Estes testes realizaram-se no rio Lizandro, situado na Ericeira. Este rio é um bom local para testar o protótipo pois possui algumas perturbações, não só devido à sua corrente, mas

também por ser um local o vento é predominante. A figura seguinte apresenta o protótipo a navegar no rio Lizandro.



Figura 41 – Rio Lizandro, local utilizado para testes ao controlador PID.

Como se pode observar é um rio espaçoso para testar o protótipo, também é um rio com alguma corrente e ligeira ondulação que é bastante sentida no protótipo devido às suas pequenas dimensões.

A figura abaixo representa o primeiro teste realizado neste local, para testar o controlo a uma curta distância.



Figura 42 – Testes no rio, a curta distância. (Google Earth).

Como se pode observar na figura acima referida, o protótipo iniciou a sua trajetória no  $Wp$ , afastou-se com a corrente e com o vento e, ao afastar-se mais de três metros da posição, liga os motores para se posicionar perto do  $Wp$ . Este teste a curta distância foi concluído com sucesso, pois o protótipo conseguiu deslocar-se para a posição desejada.

Visto o protótipo estar a fazer o que é suposto a curtas distâncias, iniciou-se uma série de testes para longas distâncias. Para o protótipo realizar testes a longas distâncias é necessário enviar-lhe as coordenadas da posição desejada. Enquanto nos testes a curtas distâncias, o controlador está programado para receber e guardar a primeira coordenada vinda do recetor GPS, como posição desejada. A figura seguinte ilustra um dos testes a longas distâncias, realizados pelo protótipo para se deslocar para o  $Wp$  (posição introduzida pelo operador, como posição desejada).

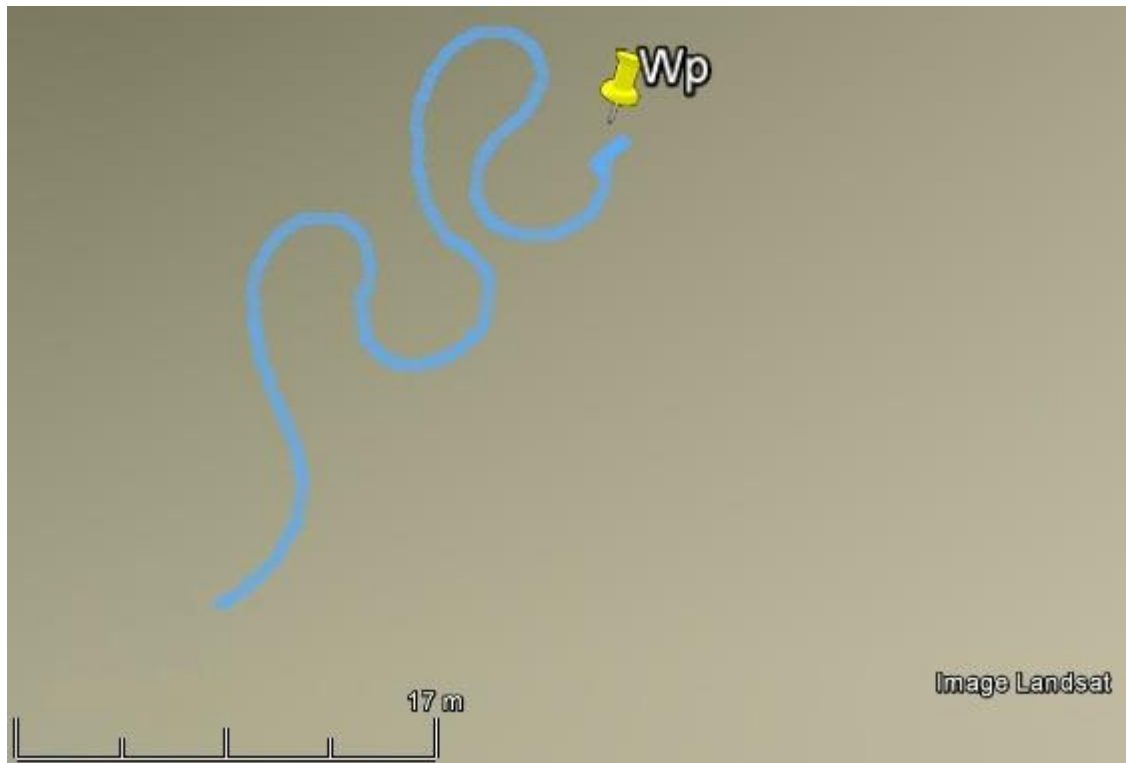


Figura 43 – Teste a longas distâncias, realizado no rio Lizandro. (Google Earth).

Ao analisar a trajetória do protótipo, pode-se dizer que concluiu o teste com sucesso. No entanto, a trajetória realizada não é a melhor. Isto é, o protótipo conseguiu deslocar-se para a posição indicada pelo operador, mas não pelo melhor trajeto.

Para melhorar essa trajetória foi melhorado o controlador PID. Essa melhoria consistiu em aumentar a constante  $Kd$ , que tinha pouca influência no controlador. Outra alteração foi alterar a potência máxima do motor de bombordo, que é mais potente que o de estibordo.

Depois destas alterações, foram efetuados mais uma variedade de testes ao controlador. A figura seguinte representa um desses testes, a longa distância da posição desejada.



Figura 44 – Teste ao controlador PID a longa distância, no rio, depois de alterações.  
(Google Earth).

Teste concluído com sucesso. Ao analisar este teste, verifica-se que o protótipo realizou uma trajetória ideal. O protótipo, inicialmente liga os dois motores à mesma velocidade, para obter uma proa aceitável, pois só em movimento é que é possível. Depois movimentasse de modo a orientar-se segundo um azimute calculado. Numa fase inicial, o controlador tem um erro maior em relação ao azimute (distância lateral em relação ao azimute calculado), que vai reduzindo ao longo da sua trajetória.

Depois de vários testes, todos com sucesso, a longas distâncias, foi realizado outra série de testes mas a curtas distâncias sem efetuar alterações à programação do controlador. A figura seguinte representa um dos testes efetuados a curta distância.

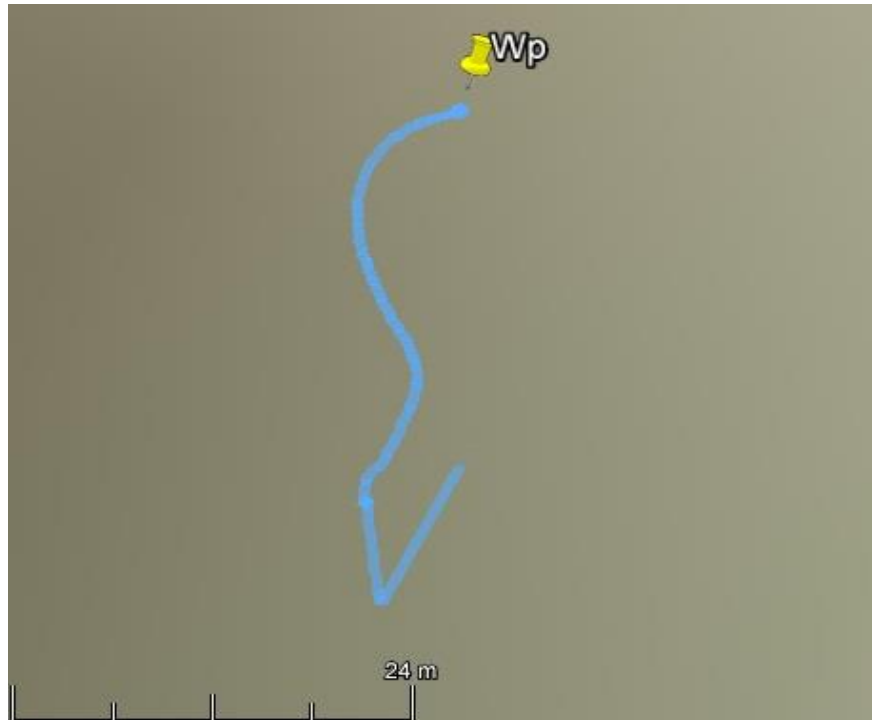


Figura 45 – Teste ao controlador PID, a curta distância. (Google Earth).

Como se pode visualizar nesta figura, o protótipo inicialmente colocar os motores à mesma velocidade para obter uma proa válida, e de seguida guina de forma a dirigir-se para a posição desejada. Em comparação com a figura do teste anterior, este tem um erro maior em relação ao azimute calculado pois tem menos distância para corrigir esse mesmo erro.

No entanto esta série de testes foi concluída com sucesso.

#### 4.1.3 CONCLUSÃO

Após realizar diversos testes em sítios diferentes e com condições diferentes, foi possível melhorar o controlador PID ao protótipo. O controlador PID é eficaz, no entanto é preciso adequar as constantes ( $K_p$ ,  $K_i$  e  $K_d$ ) consoante os motores e a deslocação da embarcação. Para adequar estas constantes, demorou algum tempo e foi necessário bastantes testes. No entanto, foi concluído com sucesso. O protótipo conseguiu tanto em águas paradas como em águas com corrente e vento, efetuar uma trajetória eficaz.

# CAPÍTULO 5 - CONCLUSÃO

---

Neste capítulo pretende-se resumir o trabalho desenvolvido ao longo do período que foi designado para o desenvolvimento desta tese de mestrado. Será feita uma revisão dos objetivos alcançados e dos que ficaram por alcançar. Pretende-se também tirar conclusões detalhadas acerca do projeto e deixar algumas sugestões para o futuro com o intuito de desenvolver e melhorar este projeto de posicionamento automático.

Inicialmente foram propostos os seguintes objetivos:

- Fazer uma seleção e realizar um estudo dos equipamentos necessários para criar um sistema de posicionamento autónomo;
- Programar um sistema de posicionamento autónomo com um controlador;
- Testar o sistema de posicionamento autónomo.

O primeiro objetivo faz referência a todo o estudo e soluções relativas ao sistema por completo. O Capítulo 2, Estado de Arte, é onde se insere esse objetivo. Neste Capítulo é feito um estudo sobre os diversos sistemas de posicionamento dinâmico, sobre algoritmos de controlo e relativamente ao *Hardware* que foi utilizado. O sistema de controlo escolhido para utilizar neste projeto foi o controlador proporcional, integral e derivativo. Face aos requisitos definidos, este é o que mais se adequa.

O segundo objetivo, relativo à programação, está descrito no Capítulo 3, Arquitetura do Sistema. Este capítulo é dividido em três partes, *Hardware*, *Software* e *Interface*. É no subcapítulo do *Software* que é descrita a programação do sistema de posicionamento. No subcapítulo *Hardware* é onde estão descritas todas as ligações físicas efetuadas. No *Interface* está apresentado como é realizada a ligação entre o sistema de controlo e o sistema de propulsão, bem como a comunicação entre eles.

O último objetivo, e não menos importante, está relacionado com os testes ao projeto. Está presente no Capítulo 4, Simulador e Testes, não só um simulador que foi utilizado para testar o algoritmo *way-point* e o algoritmo *line-tracking*, como também testes realizados em campo. Este simulador programado é demasiado simples para o pretendido. No entanto concebeu-nos uma ideia de como programar os algoritmos de posicionamento, e qual seria o mais eficaz para o mesmo cenário.

Nos testes realizados em campo, está uma sequência de testes escolhidos que representa o evoluir do sistema de posicionamento. À medida que se iam fazendo testes, o sistema de posicionamento ia sendo ajustado consoante o comportamento do projeto. Com os resultados finais, pode-se dizer que o projeto realiza uma trajetória ideal, e que cumpre com os requisitos apresentados inicialmente.

O protótipo utilizado nestes testes em campo é leve, pequeno, e fácil de transportar. Utiliza também dois motores elétricos, tal como idealizado inicialmente. No entanto, tem algumas desvantagens pelo facto de ser frágil, possuir pouco espaço para colocar todo o *Hardware* relativo ao sistema de posicionamento, ter pouca altura e assim ser pouco influenciado pelo vento e corrente. Outra grande desvantagem é o facto de ter os motores muito à superfície da água, assim o protótipo tem dificuldades em se deslocar quando existe ondulação.

O projeto tem a capacidade de ser um sistema capaz de se posicionar autonomamente, calcula a sua trajetória consoante a posição para a qual tem de se deslocar, tem rapidez de resposta e é capaz de se posicionar independentemente das condições ambientais, exceto se as condições ambientais não permitirem que o projeto navegue.

Em suma, posso concluir que é viável a utilização de um controlador deste tipo. No entanto, o controlador PID requer alguns ajustes nas suas constantes ( $K_p$ ,  $K_i$ ,  $K_d$ ), que variam consoante a estrutura da plataforma onde se insere, e consoante a resposta dos motores. Assim, embora seja viável a utilização desta solução, é necessário ajustar as constantes deste controlador ao mudar de protótipo.

## DESENVOLVIMENTOS FUTUROS

Com o desenvolvimento do projeto muitos requisitos se tornam mais evidentes. Segue abaixo mencionado os desenvolvimentos futuros que o autor acha mais relevante.

### **Protótipo:**

Desenvolver o projeto em tamanho real. Depois de concluir a viabilidade na utilização de um controlador PID, é de todo vantajoso construir um projeto em tamanho real.

### **Gestor de energia:**

Desenvolver um gestor de energia, de modo a gerir toda a energia proveniente das diversas fontes.

### **Simulador:**

Elaborar um simulador real, que em comunicação com o sistema de posicionamento (Arduíno), teste os algoritmos de posicionamento. Deste modo, seja mais fácil o ajuste das constantes do controlador PID.

### **Segurança:**

Para a segurança do projeto, é vantajoso implementar um cartão SIM, ou outra forma de comunicação para Terra, de forma a alertar caso o projeto se afaste muito da posição por algum motivo.

# REFERÊNCIAS BIBLIOGRÁFICAS

---

- Almeida, R. (2010). Sistema Inteligente de Posicionamento Acústico Subaquático. Tese de mestrado, Faculdade de Engenharia da Universidade do Porto, Porto.
- Andrade, B. (s.d.). *Apostila de Eletrônica – Analógica/Potência*. Obtido em 30 de Setembro de 2012, de <http://pt.scribd.com/doc/51068532/59/Arquitetura-de-dispositivos-FPGAs>.
- Arduino. (2006). Obtido em 23 de Setembro de 2012, de <http://www.arduino.cc>.
- Aström, K.; Wittenmark, B. (1997). *Computer-Controlled Systems*. 3rd Edition, Prentice-Hall.
- Barreiros, D. (2012). *eVentos - Sistema de Comando e Controlo GroundStation*. Tese de mestrado, Escola Naval, Alfeite.
- Bertini, L. (2004). *Microcontroladores PIC16F84 e PIC16F628*. Obtido em 30 de Setembro de 2012, de [http://www.luizbertini.net/microcontroladores/apostila\\_pic\\_16f84\\_16f628.pdf](http://www.luizbertini.net/microcontroladores/apostila_pic_16f84_16f628.pdf).
- Bray, D. (1998). *Dynamic Positioning - The Oilfield Seamanship Series*. Volume 9, Oilfield Publication Ltd.
- Buscariollo, P. (2008). *Sistema de Posicionamento Dinâmico baseado em Visão Computacional e Laser*. Tese do doutoramento, Escola Politécnica da Universidade de São Paulo, São Paulo.
- Cavaco, T. (2011). *Projecto eVentos – Sensores*. Tese de mestrado, Escola Naval, Alfeite.
- Cinel (s.d.). *Microprocessadores e microcontroladores*. Obtido em 30 de Setembro de 2012, de <http://www.cinelformacao.com/mpics/files/ud3/ud3cap1p1.htm>.
- Cooking hacks (s.d.) a. New 3G + GPS shield for Arduino. Obtido em 10 de Outubro de 2012, de <http://www.cooking-hacks.com/index.php/documentation/tutorials/arduino-3g-gprs-gsm-gps>.
- Cooking hacks (s.d.) b. Documentation: GPRS/GSM Quadband Module for Arduino Tutorial (SIM900). Obtido em 10 de Outubro de 2012, de <http://www.cooking-hacks.com/index.php/documentation/tutorials/arduino-3g-gprs-gsm-gps>.

hacks.com/index.php/documentation/tutorials/arduino-gprs-gsm-quadband-sim900.

- Farnell (2007). RF SOLUTIONS - GPS-610F – MODULE. Obtido em 22 de Setembro de 2012, de <http://pt.farnell.com/rf-solutions/gps-610f/module-gps-receiver-16ch-lo-volt/dp/1777460?ref=lookahead>.
- Fernandes, D. (2008). *Sistema de controle ótimo para veículo submersível semi-autônomo*. Tese de mestrado, Escola Politécnica da Universidade de São Paulo, São Paulo.
- Ferreira, J. (2010). *Sistema de aquisição de dados para um veículo elétrico, Dissertação de Mestrado Integrado em Engenharia Eletrotécnica e de Computadores*. Porto: Universidade do Porto, Faculdade de Engenharia.
- Fossen, T. (1994). *Guidance and Control of Ocean Vehicles*. John Wiley and Sons Ltd.
- Garcias, N., et al. (2003). *Mosaic-based navigation for autonomous underwater vehicles*. Oceanic Engineering, Volume 28, October 2003.
- Gopi, S. (2005). *Global Positioning System – Principles and Applications*. Tata McGraw-Hill.
- Graham, C.; Goodwin, S.; Graebe, M. (s.d.). *Classical PID Control*. Prentice Hall PTR.
- Guerra, W. (2009). *Implementação de Controle Proporcional, Integral e Derivativo Digital em Controladores Lógico Programáveis*. Monografia, Universidade Federal de Pernambuco, Recife.
- Kaplan, E.; Hegarty, C. (2006). *Understanding GPS – Principles and Applications*. (2ª Edição). Norwood, Ma:Artech house, Inc.
- Lopes, F.; Miguens, P.; Reis, J.; Sampaio, P. (s.d.). Análise e comparação de algoritmos de cálculo de posições GPS. Lisboa: Departamento de Engenharia Electrónica e Telecomunicações e de Computadores, ISEL.
- Lopes, P. (2012). Boia “à deriva”, faz cancelar primeira regata do dia. *Diário de Notícias*.
- Kaplan, E. D., & Hegarty, C. J. (2006). *Understanding GPS Principles and Applications*. Londres: Artech House.
- Marques, N. (2012). *eVentos - Sistema de Navegação Autônoma*. Tese de mestrado, Escola Naval, Alfeite.

- Massamba, J. (2009). *Sistema de seguimento automático do UAV*. Tese de mestrado, Escola Naval, Alfeite.
- McGovern A. (2003). *Geographic Distance and Azimuth Calculations*. Obtido em 8 de Outubro de 2012, de <http://www.codeguru.com/cpp/cpp/algorithms/article.php/c5115/Geographic-Distance-and-Azimuth-Calculations.htm>.
- Monteiro, L. (2005). *The Portuguese DGPS network*. Lisboa: Academia de Marinha.
- Morimoto, C. E. (2007). *FPGA*. Obtido em 30 de Setembro de 2012, de <http://www.hardware.com.br/termos/fpga>.
- Mundo da Náutica, (2011). Obtido em 3 de Fevereiro de 2013, de <http://www.mundodanautica.com/BOIAS/3168-boias-de-marcacao-de-regatas-.html>.
- Ogata, K. (2002). *Modern Control Engineering* (4ª edição). Azeizh.
- Palma, J. (2008). *Accionamentos Electromecânicos de Velocidade Variável*. (2ª Edição). Fundação Calouste Gulbenkian, Lisboa.
- Pessanha, N. (2006). *Arduino e Recursos Avançados*. Tutorial da cadeira de Microprocessadores, Escola Naval, Alfeite.
- Ramos, N. (n.d.). Mundo da Vela. Obtido em 14 de Janeiro de 2013, de <https://sites.google.com/site/marvelavento/lingua-language/portugues/regatas>.
- Rocha, G. C. (2011). *Projecto eVentos – Sistema de Controlo*. Tese de mestrado, Escola Naval, Alfeite.
- Roving Networks (2012). *Roving Networks Modules*. Obtido em 10 de Outubro de 2012, de [http://www.rovingnetworks.com/Roving\\_Network\\_Modules](http://www.rovingnetworks.com/Roving_Network_Modules).
- Sarmiento, M. (2008). *Guia Prático sobre a Metodologia Científica para Elaboração, Escrita e Apresentação de Teses de Doutoramento, Dissertações de Mestrado e Trabalhos de Investigação Aplicada*. Lisboa: Universidade Lusíada de Lisboa.
- Santos, F. L. C. (s. d.). *Microcontroladores PIC*. Obtido em 30 de Setembro de 2012, de <http://www.ebah.com.br/content/ABAAABWVMAI/microcontroladores-pic>.
- Santos, N. (2008). *Navegação de veículo submarino autónomo em rede acústica móvel*. Tese de mestrado, Faculdade de Engenharia da Universidade do Porto, Porto.

- Santos, N. P. (2009). *Arduino – Introdução e recursos avançados*.
- Santos, N.P. (2010). *Sistema de Telemetria Aplicado a uma Plataforma Naval*. Tese de mestrado, Escola Naval, Alfeite. pp. 76-79.
- Silva, J. (2000). *Controle em malha fechada*. Obtido em 27 de Julho de 2013, de <http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node6.html>.
- Sparkfun (2006). *Datasheet of GPS smart antenna module, LS20030~3*. Obtido em 23 de Setembro de 2012, de [http://www.sparkfun.com/datasheets/GPS/Modules/LS20030~3\\_datasheet\\_v1.2.pdf](http://www.sparkfun.com/datasheets/GPS/Modules/LS20030~3_datasheet_v1.2.pdf).
- Tannuri, E. (2009). *Sistemas de posicionamento dinâmico: projeto, análise e novos desenvolvimentos*. Tese de mestrado, Escola Politécnica da Universidade de São Paulo, São Paulo.
- Tomtom. (2012). *Erro induzido pela atmosfera & Erro multi-path*. Obtido em 22 de Novembro de 2012, de <http://www.tomtom.com/howdoesitwork/page.php?ID=22&CID=6&Language=17>.
- Veness, C. (2002). *Calculate distance, bearing and more between Latitude/Longitude points*. Obtido em 8 de Outubro de 2012, de <http://www.movable-type.co.uk/scripts/latlong.html>.
- Veríssimo, L.; Ferreira, S.; Serrasqueiro, T.; Pereira, V. (2008). *Viagens com GPS*. Obtido em 22 de Setembro de 2012, de <http://wannabescientists.blogs.sapo.pt/383.html>.

# APÊNDICE A – BIBLIOTECA DO ARDUINO

---

## **INIT\_PC:**

Rotina: void INIT\_PC (int var\_1);

Parâmetros de entrada:

- var\_1 - Velocidade da comunicação *serial* em *baud per second* (Bd).

Parâmetros de saída: Nenhum.

Descrição: Define a velocidade da comunicação da porta *serial* 0.

## **INIT\_GPS:**

Rotina: void INIT\_GPS (int var\_1);

Parâmetros de entrada:

- var\_1 - Velocidade da comunicação *serial* em *baud per second* (Bd).

Parâmetros de saída: Nenhum.

Descrição: Define a velocidade da comunicação da porta *serial* 3.

## **INIT\_SD:**

Rotina: boolean INIT\_SD ();

Parâmetros de entrada: Nenhum.

Parâmetros de saída:

- Presença ou ausência/falha de cartão SD.

Descrição: Verifica presença ou ausência/falha de cartão SD, caso o valor da variável do tipo boolean.

## **GET\_GPS:**

Rotina: void GET\_GPS ();

Parâmetros de entrada: Nenhum.

Parâmetros de saída:

- Frase NMEA com o código \$GPMRC na variável global GPS\_LOG.

Descrição: Esta rotina seleciona das várias frase fornecidas através do GPS, a que contém o código GPMRC, verifica se o GPS já está a receber posição e por fim verifica se o *checksum* é válido.

## **LOG\_FILE:**

Rotina: void GET\_LOGFILE(char var\_1[], char var\_2[]);

Parâmetros de entrada:

- var\_1 – Informação a ser gravada no cartão;
- var\_2 – Local para guardar informação.

Parâmetros de saída: Nenhum.

Descrição: Esta rotina guarda a informação da var\_1, no sítio do cartão SD indicado em var\_2.

# APÊNDICE B – ROTINAS DO ARDUINO

---

## INICIALIZAÇÕES

### INIT\_VAR:

Rotina: void INIT\_VAR ();

Parâmetros de entrada: Nenhum.

Parâmetros de saída: Nenhum.

Descrição: Declaração de variáveis iniciais para o bom funcionamento do programa, tais como:

- Inicializa uma série de variáveis de controlo de atividades como leitura de sensor efetuada (GPS\_READ), inicializa também variável que controla se a primeira gravação já foi efetuada (SAVE), tal como assegura a inicialização de variáveis que controlam os motores.

## GRAVAÇÃO DE DADOS

### VALID\_LOG:

Rotina: void VALID\_LOG ();

Parâmetros de entrada: Nenhum

Parâmetros de saída: Nenhum

Descrição: A rotina tem como objetivo verificar se o ficheiro “T00” existe no cartão SD. Se existir, a rotina vai somando uma unidade, a T00, ficando, T01, T02, e assim sucessivamente até chegar a um que não exista ou até atingir T99.

## **SAVE\_VAR:**

Rotina: void SAVE\_VAR (char var\_1 [n], char var\_2 [n], char var\_3 [n])

Parâmetros de entrada:

- var\_1 – Variável que contem a frase do GPS;
- var\_2 – Variável que contem frase da agulha;
- var\_3 – Variável na qual se guarda o conteúdo a gravar no cartão.

Parâmetros de saída: Nenhum

Descrição: A rotina tem como objetivo juntar a informação a guardar no cartão SD numa só variável.

## **CÁLCULOS DE NAVEGAÇÃO**

### **GET\_BUOY\_DIST:**

Rotina: int GET\_BUOY\_DIST (float var\_1, float var\_2, float var\_3, float var\_4)

Parâmetros de entrada:

- var\_1 – Latitude em graus decimais do ponto final para determinação da distância;
- var\_2 – Longitude em graus decimais do ponto final para determinação da distância;
- var\_3 – Latitude em graus decimais a partir do qual se quer calcular a distância (posição atual);
- var\_4 – Longitude em graus decimais a partir do qual se quer calcular a distância (posição atual).

Parâmetros de saída:

- Distância entre os dois pontos em metros.

Descrição: A rotina tem como objetivo calcular a distância entre dois pontos geográficos.

### **GET\_BTWP:**

Rotina: int GET\_BTWP (float var\_1, float var\_2, float var\_3, float var\_4);

Parâmetros de entrada:

- var\_1 – Latitude em graus decimais do ponto final para determinação do rumo;
- var\_2 – Longitude em graus decimais do ponto final para determinação do rumo;
- var\_3 – Latitude em graus decimais do ponto inicial para determinação do rumo;
- var\_4 – Longitude em graus decimais do ponto inicial para determinação do rumo.

Parâmetros de saída:

- Rumo entre os dois pontos em graus.

Descrição: A rotina tem como objetivo calcular o rumo entre os dois pontos geográficos.

## **CONTROLO DE MOTORES**

### **GET\_ERRO:**

Rotina: int GET\_ERRO (int var\_1, int var\_2);

Parâmetros de entrada:

- var\_1 – proa atual;
- var\_2 – azimute à posição desejada;

Parâmetros de saída:

- Erro para o controlo PID.

Descrição: A rotina tem como objetivo calcular o erro que o controle PID vai ter em conta.

### **GET\_DRIVE:**

Rotina: int GET\_DRIVE (int var\_1, int var\_2, int var\_3);

Parâmetros de entrada:

- var\_1 – erro atual;
- var\_2 – erro anterior;
- var\_3 – tempo total que o erro persiste.

Parâmetros de saída:

- Diferencial dos motores.

Descrição: A rotina tem como objetivo calcular o diferencial dos motores (em PWM) para reduzir o erro.

## APÊNDICE C – CHECKLIST PARA TESTES

---

Lista de verificação de material para efetuar teste:

<b>MATERIAL</b>	
Baterias	
Carregador de baterias	
GPS portátil	
Embarcação	
Microcontrolador	
Cartão de memória	
Placa de controlo	
Fita isoladora (isolar possíveis entradas de água)	
Chaves Philips (abrir embarcação)	
Toalha (secar a embarcação)	
Computador	

Lista de preparativos para efetuar teste:

<b>Preparativos</b>		
<b>6 - H</b>	Carregar baterias da embarcação	
<b>1 - H</b>	Upload do programa no microcontrolador	
	Alimentar placa de controlo	
<b>40 - H</b>	Verificar se GPS adquire sinal	
	Verificar se os motores funcionam	
<b>10 - H</b>	Fechar a embarcação	
<b>H</b>	Registo Início e Fim da série	
	Registo Temporal	
	Registo situações relevantes	

## **APÊNDICE D – CD-ROM**

---