

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



**FEUP**

# Controlo Coordenado de Veículos Autónomos Submarinos e de Superfície

José Luís da Rocha Melo

Tese submetida no âmbito

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Major de Automação

Orientador: Aníbal Matos

Maio de 2008



A Dissertação intitulada

**“Controlo coordenado de veículos autónomos submarinos e de superfície”**

foi aprovada em provas realizadas em 24/Julho/2008

o júri

Presidente



Professor Doutor António Paulo Gomes Mendes Moreira  
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto



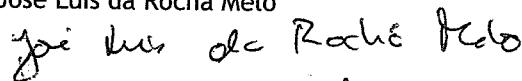
Professor Doutor Luís Miguel Pinho de Almeida  
Professor Auxiliar da Universidade de Aveiro



Professor Doutor Aníbal Castilho Coimbra de Matos  
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

Autor - José Luis da Rocha Melo



Faculdade de Engenharia da Universidade do Porto



# Resumo

No decorrer do trabalho que tem vindo a ser efectuado pelo Ocean Systems Group, um grupo de estudo associado ao Instituto de Sistemas e Robótica - Porto, verificou-se que haveria grande interesse em que Veículos Autónomos de Superfície (ASVs) e Veículos Autónomos Subaquáticos (AUVs) demonstrassem um movimento coordenado na execução de várias missões. Concretamente, seria interessante que um ASV pudesse seguir à superfície ou mesmo alcançar a trajectória realizada por um AUV, enquanto este executa uma dada missão para a qual foi previamente programado. Esta dissertação centra-se no controlo de um Veículo Autónomo de Superfície, ASV, de tal forma que a sua trajectória coincida com a trajectória feita por um AUV.

Por forma a obter o desejado movimento de coordenação entre o AUV e o ASV, faria sentido programar cada um dos veículos com as mesmas missões. Porém, devido à natureza de cada um dos veículos, das missões e também devido aos sistemas de localização usados, este tipo de solução criaria, com certeza, problemas de sincronização e temporização de movimentos, problemas esses bastante difíceis de ultrapassar.

A solução aqui proposta, composta por duas fases distintas, tenta em primeiro lugar prever a posição do AUV, controlando em seguida o ASV de tal forma que a sua trajectória seja a desejada. Com base nos sinais trocados entre o AUV e a rede acústica de localização utilizada para esse efeito, é possível estimar a posição do veículo, com recurso a um filtro de kalman, prevendo depois a sua trajectória rectilínea com o auxílio de um algoritmo de mínimos quadrados recursivos. Derivando uma lei de controlo apropriada, provada estável pelo método de estabilidade de Lyapunov, controla-se então o ASV para que a sua trajectória converja com a do AUV.

O trabalho será concluído com a apresentação de resultados experimentais que permitam avaliar o bom desempenho das soluções propostas.



# Abstract

Following the work being done by the Ocean Systems group, a research group within the Institute of Systems and Robotics - Porto, it has been proved of great interest that Autonomous Surface Vehicles and Autonomous Underwater Vehicles could present a coordinated motion on performing various activities. To be more precise, it would be interesting that an ASV could follow, at the surface, trajectory performed by an AUV while this executes a previously programmed mission. This thesis is focused on the control of an ASV in such a way that its motion matches the trajectory of an AUV.

In order to accomplish this coordinated movement, between AUV and ASV, it would make sense to just program both with the same missions. However, not only due to the nature of the missions, as well as due to the nature of the vehicles and their navigational systems, by choosing this kind of approach, some problems difficult to overcome would arise, like synchronization and timing problems.

Therefore, the solution hereby proposed tries, on a first stage, to predict the position of the AUV, and then to actuate the ASV in accordance so that the trajectories of both vehicles match. By using information about the acoustic signals exchanged between the vehicle and the acoustic network that supports the operation, it is possible to initially estimate the current AUV position, and to predict its rectilinear trajectory with a least squares algorithm. Then, and by using a proper control law, proved stable by the Lyapunov stability method, the ASV will be controlled in accordance, so that its trajectory meets the desired one.

The thesis will end with the presentation of experimental results that allow to evaluate the overall performance of the proposed solutions.





# Agradecimentos

Não poderia deixar de agradecer ao meu orientador, Dr. Aníbal Matos, por todo o apoio e confiança demonstrados ao longo de todo o semestre, e que possibilitaram que este trabalho fosse concluído com sucesso.

Também a todos aqueles que tão importantes para mim foram ao longo de todo o curso, por todo o apoio, por toda a compreensão, por todos os momentos, um sincero obrigado. Em especial ao Tomé Costa, à Diana Costa, ao Tiago Nunes, ao Vítor Torres, ao José Barbosa, ao José Araújo e ao Nuno Medeiros, não só pelo companheirismo, mas especialmente pela amizade.

José Melo



*Aos meus pais. Por tudo.*



# Conteúdo

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introdução</b>                                  | <b>1</b>  |
| 1.1      | Motivação e contexto do trabalho . . . . .         | 1         |
| 1.2      | Breve especificação do problema . . . . .          | 1         |
| 1.3      | Objectivos do trabalho . . . . .                   | 2         |
| 1.4      | Estrutura do Relatório . . . . .                   | 2         |
| <b>2</b> | <b>Estado da Arte</b>                              | <b>3</b>  |
| 2.1      | Modelos do Sistema . . . . .                       | 3         |
| 2.2      | Navegação e Posicionamento . . . . .               | 6         |
| 2.2.1    | Métodos de Posicionamento Acústico . . . . .       | 6         |
| 2.3      | Filtro de Kalman . . . . .                         | 8         |
| 2.4      | Controlo não linear . . . . .                      | 10        |
| 2.4.1    | Estabilidade de Lyapunov . . . . .                 | 11        |
| 2.4.2    | Funções de Lyapunov . . . . .                      | 11        |
| 2.5      | Síntese . . . . .                                  | 12        |
| <b>3</b> | <b>Veículos e Estrutura Computacional</b>          | <b>15</b> |
| 3.1      | AUV MARES . . . . .                                | 15        |
| 3.2      | ASV Zarco . . . . .                                | 17        |
| 3.3      | Bóias . . . . .                                    | 17        |
| 3.4      | Modelo de Comunicações . . . . .                   | 18        |
| 3.5      | Repositório . . . . .                              | 19        |
| <b>4</b> | <b>Algoritmos Desenvolvidos</b>                    | <b>21</b> |
| 4.1      | Estimação da Posição do AUV . . . . .              | 21        |
| 4.1.1    | Algoritmo de Estimação . . . . .                   | 22        |
| 4.1.2    | filtro de Kalman . . . . .                         | 24        |
| 4.1.3    | Estimador Recursivo de Mínimos Quadrados . . . . . | 27        |
| 4.2      | Guidance do ASV . . . . .                          | 29        |
| <b>5</b> | <b>Controlo do ASV</b>                             | <b>31</b> |
| 5.1      | Controlo de Heading . . . . .                      | 31        |
| 5.1.1    | Estabilidade do Controlo . . . . .                 | 34        |
| 5.1.1.1  | Caso 1 . . . . .                                   | 35        |
| 5.1.1.2  | Caso 2 . . . . .                                   | 36        |
| 5.1.1.3  | Caso 3 . . . . .                                   | 36        |
| 5.1.1.4  | Restantes . . . . .                                | 37        |
| 5.1.2    | Controlador . . . . .                              | 37        |

|          |  |           |
|----------|--|-----------|
| 5.1.3    | Actuação do Veículo . . . . .                  | 38        |
| 5.2      | Controlo de Velocidade . . . . .               | 38        |
| <b>6</b> | <b>Resultados Obtidos</b>                      | <b>41</b> |
| 6.1      | Simulação . . . . .                            | 41        |
| 6.1.1    | filtro de Kalman . . . . .                     | 42        |
| 6.1.2    | Estimador de Mínimos Quadrados . . . . .       | 45        |
| 6.1.3    | Algoritmo de Guidance . . . . .                | 46        |
| 6.1.4    | Seguimento de uma recta . . . . .              | 49        |
| 6.2      | Testes em Ambiente Real . . . . .              | 49        |
| 6.2.1    | Algoritmo Estimador . . . . .                  | 52        |
| <b>7</b> | <b>Conclusões</b>                              | <b>57</b> |
| 7.1      | Resultados . . . . .                           | 57        |
| 7.2      | Trabalho Futuro . . . . .                      | 58        |
| <b>A</b> | <b>Implementação Computacional</b>             | <b>59</b> |
| A.1      | Fluxograma Algoritmo Estimador . . . . .       | 59        |
| A.2      | Fluxograma Algoritmo <i>Guidance</i> . . . . . | 59        |
| A.3      | Sockets . . . . .                              | 59        |
| A.4      | Threads . . . . .                              | 62        |
| <b>B</b> | <b>Artigo Submetido</b>                        | <b>63</b> |
| B.1      | Abstract . . . . .                             | 63        |
| B.2      | Introduction . . . . .                         | 63        |
| B.3      | Estimation . . . . .                           | 64        |
| B.4      | Control and Actuation . . . . .                | 64        |
|          | <b>Referências</b>                             | <b>68</b> |

# Lista de Figuras

|      |  |    |
|------|--|----|
| 3.1  | Integração dos algoritmos na estrutura computacional existente . . . . .   | 16 |
| 3.2  | AUV MARES . . . . .  | 16 |
| 3.3  | ASV Zarco . . . . .  | 17 |
| 3.4  | Farol Acústico . . . . .   | 18 |
| 4.1  | Troca de sinais acústicos: esquema . . . . .   | 22 |
| 4.2  | Esquematisação de uma situação de missão típica . . . . .  | 23 |
| 4.3  | Fluxograma das máquinas de estados de cada uma das bóias . . . . .   | 25 |
| 4.4  | Diagrama de blocos para a estimação da posição do AUV . . . . .  | 29 |
| 5.1  | Exemplo de comportamento desejado para o ASV . . . . .   | 33 |
| 5.2  | Controlador de Velocidade do ASV . . . . .   | 39 |
| 6.1  | Trajectória simulada do AUV (a fino) e trajectória estimada pelo filtro de Kalman (a grosso) . . . . .   | 42 |
| 6.2  | Evolução das posições simuladas (a fino) e estimadas pelo filtro de Kalman (a grosso) do AUV ao longo do eixo $x$ . . . . .  | 43 |
| 6.3  | Evolução das posições simuladas (a fino) e estimadas pelo filtro de Kalman (a grosso) do AUV ao longo do eixo $y$ . . . . .  | 43 |
| 6.4  | Evolução das velocidades simuladas (a fino) e estimadas pelo filtro de Kalman (a grosso) do AUV ao longo do eixo $x$ . . . . .   | 44 |
| 6.5  | Evolução das velocidades simuladas (a fino) e estimadas pelo filtro de Kalman (a grosso) do AUV ao longo do eixo $y$ . . . . .   | 44 |
| 6.6  | Evolução do erro de posição para valores de $\lambda$ de 0.51 (esquerda em cima), 0.80 (direita em cima), 0.89 (esquerda em baixo) e 0.93 (direita em baixo) . . . . . | 46 |
| 6.7  | Evolução das posições simuladas (a fino) e estimadas pelos Mínimos Quadrados (a grosso) do AUV. À direita $x$ e à esquerda $y$ . . . . .                               | 47 |
| 6.8  | Evolução das velocidades simuladas (a fino) e estimadas pelos Mínimos Quadrados (a grosso) do AUV. À direita $V_x$ e à esquerda $V_y$ . . . . .                        | 47 |
| 6.9  | Trajectória estimada (a fino) e descrita pelo ASV (a grosso) . . . . .   | 48 |
| 6.10 | Trajectória estimada (a fino) e descrita pelo ASV (a grosso) . . . . .   | 48 |
| 6.11 | Erro de posição em $x$ (a cheio) e em $y$ (a tracejado) ao longo do tempo . . . . .  | 48 |
| 6.12 | Erro total de posição ao longo do tempo . . . . .  | 49 |
| 6.13 | Trajectória do AUV (a cheio) e trajectória do ASV (a tracejado) para uma trajectória rectilínea e uniforme . . . . .   | 50 |
| 6.14 | Posições do AUV (a cheio) e do ASV (a tracejado) ao longo do tempo, relativas ao eixo $X$ para uma trajectória rectilínea e uniforme . . . . .                         | 50 |
| 6.15 | Posições do AUV (a cheio) e do ASV (a tracejado) ao longo do tempo, relativas ao eixo $Y$ para uma trajectória rectilínea e uniforme . . . . .                         | 51 |

|      |  |    |
|------|--|----|
| 6.16 | Trajectória do AUV: sistema de navegação do AUV (a fino) e algoritmo estimador (a cheio) . . . . .                     | 52 |
| 6.17 | Estimativas do AUV dada pelo filtro de Kalman (a fino) e dada pelo estimador de Mínimos Quadrados (a grosso) . . . . . | 53 |
| 6.18 | Trajectória do AUV: sistema de navegação do AUV (a fino) e algoritmo estimador (a cheio) . . . . .                     | 54 |
| 6.19 | Evolução da posição segundo o eixo $X$ ao longo do tempo . . . . .   | 54 |
| 6.20 | Evolução da posição segundo o eixo $Y$ ao longo do tempo . . . . .   | 55 |
| 6.21 | Trajectória do AUV: filtro de Kalman(a fino) e estimador de Mínimos Quadrados (a cheio) . . . . .                      | 56 |
| 6.22 | Evolução das velocidades ao longo do eixo $X$ (à esquerda) e $Y$ (à direita) ao longo do tempo . . . . .               | 56 |
| A.1  | Fluxograma do algoritmo Estimador . . . . .  | 60 |
| A.2  | Fluxograma do algoritmo de <i>Guidance</i> . . . . .   | 60 |



# Lista de Tabelas

|     |  |    |
|-----|--|----|
| 2.1 | Notação Utilizada . . . . .                  | 4  |
| 3.1 | Estrutura das Mensagens Trocadas . . . . .   | 19 |
| 4.1 | Ciclo de Comunicações . . . . .              | 23 |
| 6.1 | Estimativa de valores de $\lambda$ . . . . . | 45 |



# Capítulo 1

## Introdução

### 1.1 Motivação e contexto do trabalho

O Ocean Systems Group (OSG), um grupo de trabalho associado ao Instituto de Sistemas e Robótica - Porto, da Faculdade de Engenharia da Universidade do Porto, (FEUP), tem já um extenso trabalho realizado no âmbito dos sistemas oceanográficos. Os seus projectos de Investigação e Desenvolvimento incidem fundamentalmente no desenvolvimento de sistemas avançados de recolha e processamento de dados em ambientes marinhos.

Para a realização dos vários trabalhos de investigação com que se depara, o OSG dispõe já de soluções inteiramente desenvolvidas nos seus laboratórios e que passam por veículos aquáticos autónomos capazes de levar a cabo um conjunto de missões programadas.

De entre os veículos que dispõe, salienta-se o MARES, um veículo autónomo subaquático (Autonomous Underwater Vehicle - AUV) modular e robusto, capaz de operar em águas pouco profundas e o Zarco, um veículo autónomo de superfície (Autonomous Surface Vehicle), de pequena dimensão, e do tipo *catamaran*, especialmente concebido para operar em águas calmas.

Destaca-se ainda a existência de uma rede acústica, constituída por várias bóias ou faróis acústicos, que permite a operação destes veículos e a coordenação entre os seus movimentos.

### 1.2 Breve especificação do problema

Nas várias missões executadas pelo AUV MARES, levadas a cabo no âmbito dos diversos projectos que o OSG vai desenvolvendo, surgiu a necessidade de que a trajectória do AUV fosse seguida pelo ASV, à superfície, acompanhando inclusive o movimento do primeiro.

Para atingir este movimento coordenado entre os dois veículos, seria natural que se optasse por pré-programar ambos com a mesma missão. No entanto, atingir a desejada coordenação entre os veículos optando por esta via seria, porventura, pouco provável. De

facto, dada a natureza dos veículos e das missões, seria necessário lidar com problemas de temporização e sincronização inerentes a uma solução deste tipo.

Por outro lado, a solução alternativa de tentar estimar a trajectória do AUV, actuando o ASV em conformidade, para além de atingir objectivos puramente académicos bem mais aliciantes, trará ainda benefícios de desempenho significativos para o grupo, já que permitirá estruturar as missões que presentemente desenvolve para objectivos mais ambiciosos. Desta forma, será possível o ASV seguir com algum grau de precisão as trajectórias executadas pelo AUV, mesmo que estas sejam, à partida, desconhecidas.

### 1.3 Objectivos do trabalho

O objectivo deste trabalho será desenvolver os algoritmos necessários para que seja possível um controlo coordenado entre o AUV e o ASV, de tal forma que o último consiga seguir a trajectória do primeiro. Para isso será necessário:

- Estimar a posição do AUV em tempo real
- Estabelecer uma estratégia de controlo para o ASV que garanta a estabilidade do mesmo
- Estimar as grandezas de controlo de tal forma que permitam que o ASV siga uma trajectória adequada

O OSG tem já uma estrutura computacional bem definida bem como um modelo de comunicações específico. Dessa forma, todo o trabalho desenvolvido deverá ter em consideração uma integração com os sistemas existentes.

Os algoritmos desenvolvidos deverão ainda passar por uma fase de teste intensiva, com recurso aos simuladores apropriados, já existentes. No final, pretende-se apresentar ainda resultados obtidos em ambiente real, que permitam atestar o desempenho de todo o trabalho.

### 1.4 Estrutura do Relatório

Nos capítulos seguintes vai sendo apresentado todo o trabalho elaborado. No capítulo 2 é apresentado algum *estado da arte* no que diz respeito à navegação acústica. São também introduzidos alguns conceitos teóricos, que foram necessários para o desenvolvimento deste trabalho. No capítulo 3, é apresentada toda a estrutura do Ocean Systems Group, e que vai desde os veículos em si, até ao modelo de comunicações estabelecido. No capítulo 4 são apresentados os algoritmos desenvolvidos e é discutido o seu funcionamento. No capítulo 5 discutem-se as estratégias de controlo adoptadas enquanto que no capítulo 6 são apresentados os resultados obtidos. A conclusão e a discussão de trabalho futuro são abordados no capítulo 7.

## Capítulo 2

# Estado da Arte

Neste capítulo são introduzidos alguns conceitos teóricos necessários à abordagem do problema a tratar. Aqui são agregados conteúdos bastante diversos, mas que foram fundamentais para uma melhor compreensão do problema e consequente implementação de soluções para ele.

Inicialmente foi necessário uma melhor compreensão dos modelos de ASV utilizados pelo OSG. Este estudo permitiu também ter um primeiro contacto com as equações que modelizam a dinâmica de tais veículos.

Foi também necessário compreender os processos e algoritmos que determinam a navegação dos vários veículos do OSG, e que integram uma rede acústica. Sendo assim, também são apresentados conceitos que permitem uma melhor compreensão de métodos e processos de navegação acústica.

No que diz respeito à estimação do estado do veículo, foi necessário compreender as bases teóricas e funcionamento do Filtro de Kalman, um algoritmo fundamental neste trabalho.

Por último, alguma teoria de sistemas não lineares, com destaque para os conceitos de estabilidade de um sistema segundo Lyapunov, foi fundamental para teoricamente prever o comportamento do nosso sistema.

### 2.1 Modelos do Sistema

A existência de um modelo para qualquer sistema que se pretende controlar é crucial, já que permite um conhecimento profundo do processo a ser controlado, bem como das suas limitações, permitindo uma avaliação mais rigorosa dos desempenhos do sistema.

No que se refere a veículos marinhos, existe um grande estudo relativamente à modelização de barcos e embarcações de grande porte. Este estudo, em grande parte desenvolvido por escolas escandinavas, tem por base o grande desenvolvimento dos países do norte da Europa neste ramo da indústria.

É bastante comum considerar um veículo marinho como um corpo rígido, admitindo que este é indeformável e que não se criam forças entre os seus elementos individuais. Por outro lado, desprezando forças que possam existir provocadas pelo movimento relativo da terra em relação a um sistema estelar, é então possível admitir a existência de um referencial inercial na terra [1].

Em robótica aquática é comum usar dois referenciais distintos para descrever o movimento dos veículos e derivar a sua dinâmica. Em primeiro lugar define-se um referencial inercial, desprezando-se para isso acelerações. Este referencial deve usar um sistema de coordenadas do tipo NED (North - East - Down) significando isto que o eixo abcissas aponta para Norte, o eixo das ordenadas aponta para Este e o eixo das alturas/cotas aponta para baixo. Deve-se ainda definir um referencial móvel, com origem coincidente com o centro de gravidade do veículo, com as abcissas orientadas segundo o eixo longitudinal da embarcação, da ré para a popa, sendo as ordenadas segundo o eixo transversal, com direcção para estibordo, e as cotas, o eixo normal, orientado do topo para o fundo do veículo [2].

Enquanto o referencial inercial é utilizado para aplicações de localização e *guidance*, o referencial móvel, solidário com o veículo é útil para derivar a dinâmica do veículo e as suas equações do movimento, já que permite tirar partido de propriedades geométricas do próprio veículo.

Será importante nesta altura também referir a notação que é usada normalmente em aplicações de robótica, sendo também aquela que será aqui utilizada. Na tabela 2.1 podemos ver as relações entre as forças e momentos que actuam sobre cada um dos eixos do referencial girante escolhido, bem como a sua notação, utilizada doravante.

Tabela 2.1: Notação Utilizada

| DOF |                              | Forças e Momentos | Velocidades linear e angular | Posição e Ângulos de Euler |
|-----|------------------------------|-------------------|------------------------------|----------------------------|
| 1   | trans. na direcção x (surge) | X                 | u                            | x                          |
| 2   | trans. na direcção y (sway)  | Y                 | v                            | y                          |
| 3   | trans. na direcção z (heave) | Z                 | w                            | z                          |
| 4   | rot. em torno de x (roll)    | K                 | p                            | $\phi$                     |
| 5   | rot. em torno de y (pitch)   | M                 | q                            | $\theta$                   |
| 6   | rot. em torno de z (yaw)     | N                 | r                            | $\psi$                     |

Para descrever completamente o movimento de um corpo, é necessário saber determinar a sua posição e a sua orientação relativamente a um referencial. Dessa forma qualquer modelo que pretenda representar precisamente o movimento de um corpo terá 6 graus de liberdade, 3 relativos à posição e movimentos de translação, e os restantes 3 relativos à orientação e rotação do corpo.

Com base na 2ª lei de Newton e nos Axiomas de Euler, foi possível estabelecer um modelo com 6 graus de liberdade para a dinâmica de um corpo rígido, e que se pode ver na equação (2.1).

$$M_{RB}\dot{\nu} + C_{RB}(\nu)\nu = \tau_{RB} \quad (2.1)$$

Neste formato compacto,  $M_{RB}$  representa a matriz de inércia do corpo rígido e  $C_{RB}$  é a matriz das forças de Coriolis e centrípeta, provocadas pela existência de um referencial girante. Por outro lado  $\nu = [u, v, w, p, q, r]^T$  é o vector das velocidades lineares e angulares.  $\tau_{RB}$  representa o vector de todas as forças externas e momentos aplicados ao corpo rígido. Admitindo a validade do princípio de sobreposição, é então possível escrever (2.2) [3].

$$\tau_{RB} = \tau_H + \tau_{ondas} + \tau_{vento} + \tau_{correntes} + \tau_{controlo} \quad (2.2)$$

Nesta equação, que representa o vector generalizado de todas as forças e momentos aplicados ao nosso corpo,  $\tau_H$  representa as forças hidrodinâmicas que nele actuam. Entre estas contam-se as forças de massa e inércia adicionadas, e de amortecimento hidrodinâmico, entre outras. Por outro lado,  $\tau_{controlo}$  representa as forças criadas pela actuação dos motores do barco.

O modelo em (2.1) é bastante completo, descrevendo de forma bastante precisa a dinâmica de qualquer corpo rígido. Contudo, tendo em conta algumas particularidades da aplicação em questão, pode ser simplificado. Apesar da sua complexidade, é possível, com o auxílio de modelos para cada uma destas forças hidrodinâmicas, chegar a um sistema de equações não lineares que modelizam um corpo rígido.

Esse modelo, é constituído por largas dezenas de coeficientes, também chamados de derivadas hidrodinâmicas, tornando-se virtualmente impossível de resolver sem tirar partido de propriedades físicas que permitam anular alguns deles [1]. De facto, considerando as características geométricas existentes em grande parte dos veículos aquáticos, com eixos de simetria nos vários planos vem simplificar de sobremaneira o cálculo da matriz de inércia do corpo.

Por outro lado, assumindo uma distribuição de massa homogénea e fazendo a origem do referencial coincidir com o centro de massa do veículo também vem simplificar alguns cálculos. Finalmente admite-se ainda que *heave*, *roll* e *pitch* podem ser desprezados uma vez que estamos perante um movimento do veículo no plano horizontal, já que, neste caso concreto, o veículo em questão é um barco.

Considerando todas estas simplificações, é possível então obter um sistema de equações mais simples, com apenas 3 graus de liberdade, e que sintetiza o movimento de um veículo marinho de superfície. A dedução destas equações pode ser encontrada com mais detalhe em [1].

$$X = m(\dot{u} - vr) \quad (2.3)$$

$$Y = m(\dot{v} - ur) \quad (2.4)$$

$$N = I_z \dot{r} \quad (2.5)$$

Como se pode constatar pelas equações (2.3), (2.4) e (2.5), estas representam o movimento de um veículo marinho de forma totalmente desacoplada. Desta forma é possível separar as equações de velocidade, (2.3) e (2.4), da equação que define a direcção do veículo, (2.5) simplificando desta forma o controlo do veículo.

## 2.2 Navegação e Posicionamento

Uma necessidade decorrente do crescente uso de veículos autónomos, quer sejam de superfície, aéreos ou ainda aquáticos, é a determinação da sua localização. Dentro do contexto dos veículos autónomos aquáticos, onde se insere este trabalho, a determinação da localização de veículos autónomos tem algumas particularidades.

O recurso ao GPS, um sistema de localização global baseado numa rede de satélites, é bastante comum quando se fala em posicionamento e localização de veículos autónomos. No entanto, e dada a especificidade dos veículos aquáticos e do meio onde se deslocam, especialmente dos veículos subaquáticos, leva a que muitas vezes o recurso ao sistema GPS não seja viável, já que existem dificuldades na propagação das ondas electromagnéticas nestes meios físicos.

Desta forma, é habitual o recurso a redes acústicas, aproveitando as boas propriedades de transmissão de ondas acústicas proporcionadas por um meio de transmissão como a água. As redes acústicas, em conjunto com sistemas de *dead-reckoning*, quer constituídos por acelerómetros e giroscópios, quer constituídos por sonares, ou outros sistemas semelhantes, permitem já alcançar resultados bastante satisfatórios [4].

O princípio de funcionamento da navegação acústica é bastante simples: os veículos autónomos trocam sinais acústicos com cada uma das bóias ou faróis acústicos que constituem a rede, e que se encontram numa localização pré-definida. Com base nestas trocas de sinais é então possível calcular ângulos e distâncias entre AUV e cada um dos faróis acústicos utilizados, num processo que se designa por navegação esférica [5].

Alternativamente poder-se-ia utilizar o processo de navegação hiperbólica. A única diferença ao nível do cálculo da posição prende-se com o facto de se assumir que as distâncias medidas entre veículo e faróis restringem a posição do primeiro não a superfícies esféricas, mas sim a hiperbolóides, resultantes da medida de diferenças de tempos de propagação de sinais acústicos. Este processo de navegação tem a desvantagem de não permitir a localização externa do veículo, uma vez que neste caso o veículo se limita a ouvir sons emitidos pelas bóias.

### 2.2.1 Métodos de Posicionamento Acústico

Para a navegação acústica existem diversos métodos distintos, métodos esses que diferem fundamentalmente na geometria e no número de faróis acústicos utilizados. Destes métodos, o LBL (*Long Baseline*) e o SBL (*Short Baseline*) são os mais frequentemente



utilizados, já que são os que atingem desempenhos mais satisfatórios. No entanto, cada um destes dois métodos tem aplicações distintas.

O método de posicionamento acústico LBL, apesar de ser um sistema mais complexo, necessitando também de um maior número de faróis acústicos, é também o mais adequado para embarcações de pequenas dimensões, permitindo a medição de distâncias entre o veículo e cada um dos faróis acústicos. Este é também o sistema mais frequentemente utilizado, pois permite atingir precisões bastante elevadas, inferiores a 1m para frequências médias (18KHz e 36KHz), e independentes da profundidade de operação [6].

O LBL envolve o uso de um conjunto de faróis acústicos com uma posição fixa e pré-definida, e ainda um *transceiver* acoplado à embarcação cuja posição se pretende saber. Para a determinação da posição, a embarcação interroga cada uma das bóias com um sinal acústico, numa determinada frequência específica para cada boia. Por sua vez, cada um dos faróis acústicos, sempre que detecta um sinal na gama de frequências para o qual foi previamente programado, envia um sinal de resposta para a embarcação. Temporizando cada um destes eventos acústicos, é então possível determinar o tempo decorrido entre a interrogação a cada uma das bóias e a recepção do sinal de resposta respectivo. Conhecendo a velocidade das ondas acústicas na água, é então possível calcular a distância em linha recta a cada uma das bóias.

Para se obter um posicionamento sem ambiguidades em todas as três dimensões de interesse, é necessário o uso de três faróis acústicos distintos. No entanto, é aconselhável ainda o uso de um farol acústico extra, de modo a obter redundância nas medidas obtidas e, estatisticamente, um posicionamento com mais qualidade.

No entanto, o uso de vários faróis acústicos acarreta várias dificuldades, não só porque cada farol representa um investimento extra que é necessário fazer, mas também porque o uso de vários destes equipamentos complica a preparação de cada uma das missões de sobremaneira. Dessa forma, é muito comum que se pretenda reduzir o número de faróis utilizados.

O número mínimo de faróis a utilizar é de dois, sendo adicionalmente necessário o uso de qualquer tipo de equipamento que permita determinar a profundidade a que a embarcação navega. Desta forma, existe uma ambiguidade em cada par de medidas, já que se desconhece de que lado da *baseline* se encontra a embarcação. Este problema pode ser rapidamente resolvido se se garantir à partida que para cada operação as embarcações só circulam de um dos lados da *baseline*.

Apesar de o SBL ser um método bastante simples, não só ao nível dos recursos utilizados, já que requer o uso de apenas um farol acústico, como também ao nível implementação, é mais adequado para embarcações de grandes dimensões, já que necessita que cada embarcação tenha no mínimo três *transceivers* colocados no seu casco. Ao conjugar o uso de no mínimo três transdutores, este método permite ainda o cálculo de ângulos entre veículo e cada um dos faróis acústicos.

Baseados nestes dois métodos de navegação foram ainda desenvolvidos alguns métodos como o USBL (*Ultra-short Baseline*) e o LUSBL (*Long and Ultra-short Baseline*), que introduzem algumas alterações aos métodos originais procurando melhorar os seus desempenhos.

Paralelamente existem ainda numerosos trabalhos sobre o assunto, e que incluem temas tão vastos como navegação LBL sem conhecimento prévio da localização dos faróis acústicos [7], redes acústicas móveis [8], [9], ou ainda o aumento do desempenho dos métodos tradicionais de navegação acústica [10].

## 2.3 Filtro de Kalman

O problema da estimação da posição do AUV está subjacente à obtenção de medidas, que se pretendem fiáveis, de tal modo que a estimativa da sua posição seja o mais precisa possível. No entanto, e tal como sempre acontece, também as medidas de distância entre AUV e cada uma das bóias, obtidas por intermédio da troca de sinais acústicos entre estas entidades, está sujeita a erros. Dependendo da sua natureza e da sua grandeza, estes erros inerentes à obtenção de medidas poderá eventualmente afectar significativamente os resultados pretendidos. Desta forma, é com alguma naturalidade que surge a necessidade de filtragem dos dados obtidos, de modo a que sejam obtidas as melhores estimativas da posição do AUV.

Para uma filtragem dos dados obtidos perfilam-se várias opções, como filtros de mínimos quadrados, filtros de partículas, filtros de Wiener, entre outros. Contudo, em aplicações de navegação robótica são os filtros de Kalman que recebem maior destaque.

Tendo a sua génese em 1960, o filtro de Kalman é estimador recursivo que, tendo por base um conjunto de de medidas sujeitas a ruído, consegue obter uma estimativa do estado de um sistema dinâmico discreto, minimizando ainda a média do quadrado do erro. É ainda estatisticamente um estimador óptimo na perspectiva em que devolve sempre a melhor solução que minimiza o quadrado dos erros. No entanto, esta optimalidade do filtro está sujeita a algumas suposições, nomeadamente no que diz respeito ao ruído das medições, que se assume ser branco e Gaussiano. Para que o filtro convirja é também necessário que se disponha de uma boa estimativa inicial do estado [11].

Uma das razões para a grande popularidade dos Filtros de Kalman nas áreas da robótica prende-se com o facto de o Filtro de Kalman ser um estimador óptimo do estado de sistemas dinâmicos. Contudo, o facto de Filtro de Kalman ser um estimador recursivo, é também uma grande vantagem já que facilita de sobremaneira a sua implementação, uma vez que não é necessário armazenar grandes quantidades de informação para obter resultados.

De um ponto de vista mais geral, o filtro de Kalman funciona em duas etapas distintas, tendo para isso dois conjuntos de equações distintas: previsão do estado e correcção do estado.

Numa primeira fase, o filtro de Kalman estima *a priori* o estado por intermédio da dinâmica conhecida do sistema (2.6) e a sua covariância do erro (2.7), tendo por base os estados anteriores. Nestas equações, a matriz  $Q$  representa a covariância do ruído da evolução do sistema.

$$x_k^- = Ax_{k-1} + Bu_{k-1} \quad (2.6)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2.7)$$

Posteriormente, e sempre que se dispõe de uma nova medida de estado, estas estimativas são actualizadas e corrigidas. São as equações de *Measurement Update*. Em primeiro lugar é necessário calcular a matriz de ganho do sistema (2.8) e depois actualiza-se e corrige-se as estimativas do estado do sistema (2.9) e da covariância do erro (2.10), com novo conjunto de medidas. A matriz  $R$  está associada à covariância do ruído do processo de obtenção de novas medidas. A matriz  $H$  relaciona as medidas efectuadas com o estado do sistema, isto é,  $z = H(x)$ .

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (2.8)$$

$$x_k = x_k^- + K_k(z_k - Hx_k^-) \quad (2.9)$$

$$P_k = (I - K_k H)P_k^- \quad (2.10)$$

O filtro de Kalman foi inicialmente pensado para sistemas dinâmicos lineares, isto é, sistemas cuja dinâmica é descrita com recurso a um conjunto de equações diferenciais lineares. Contudo, muitas das vezes os sistemas a tratar são sistemas dinâmicos não lineares. Na prática, isto significa que para estes casos o conjunto de equações acima apresentado terá que sofrer pequenas alterações. A dinâmica do sistema é aproximada por intermédio das derivadas parciais, truncando a expansão da série de Taylor aos termos lineares. Adicionalmente, e sempre que se justifique, há ainda a possibilidade de se utilizar termos de maior ordem para estas aproximações, cometendo erros de linearização mais pequenos.

Existe portanto uma adaptação do filtro de Kalman para sistemas dinâmicos descritos por sistemas de equações não lineares, e que se denomina de filtro de Kalman estendido. O conjunto de equações que sintetiza o filtro de Kalman estendido é bastante semelhante com o original havendo apenas ligeiras diferenças. Essas diferenças prendem-se com o uso de derivadas parciais para aproximar as não linearidades existentes na dinâmica do sistema (2.11).

$$H = \frac{\partial z_k}{\partial X} \quad (2.11)$$

De salientar que o uso do Filtro de Kalman Estendido (EKF) necessita invariavelmente de uma boa estimativa inicial de uma solução para o sistema. De facto, este é um requisito para os EKF já que para estes não há garantia de optimalidade ou convergência.

## 2.4 Controlo não linear

Por sistema não linear entende-se um sistema cuja dinâmica seja descrita por um conjunto de equações diferenciais não lineares (2.12).

$$\dot{x} = f(x, t) \quad (2.12)$$

onde  $f$  é uma função não linear e  $x$  é o vector de estados. Se em particular  $f$  não depender explicitamente do tempo, então o sistema diz-se ainda autónomo ou invariante no tempo (2.13).

$$\dot{x} = f(x) \quad (2.13)$$

Os sistemas lineares e invariantes no tempo, são de facto bem mais vantajosos do ponto de vista do controlo, na medida em que são bem conhecidas os métodos de análise e as ferramentas necessárias à síntese de controladores deste tipo. Apesar disso, a modelação de sistemas físicos dinâmicos resulta na maioria das vezes em sistemas dinâmicos não lineares. A partir destes modelos complexos obtem-se invariavelmente modelos mais simples, que resultam da linearização desses sistemas em torno de pontos de funcionamento específicos. É necessário portanto abdicar de muitas das características do comportamento dinâmico. Desse modo, modelos não-lineares de qualquer sistema são, regra geral, bem mais próximos da realidade.

O recurso a modelos não lineares que descrevam o movimento do ASV é, por si só, impeditivo no que diz respeito ao uso das técnicas de controlo linear. Foi portanto necessário, um estudo aprofundado de algumas técnicas de controlo não linear. Os sistemas não lineares têm um conjunto de propriedades que os distinguem dos sistemas lineares e que não permitem que as mesmas técnicas de análise sejam utilizadas em ambos os casos. Em concreto, propriedades como por exemplo o princípio da sobreposição ou ainda a existência de um único ponto de equilíbrio não se verificam.

Existem várias técnicas para a análise de sistemas não lineares, entre os quais se destacam, pelo seu uso recorrente, o método da função descritiva, o método dos planos de fase, ou ainda a estabilidade de Lyapunov. De destacar ainda o método da linearização de Lyapunov, que se baseia na linearização dos modelos, aproximando sistemas não lineares por sistemas lineares.

### 2.4.1 Estabilidade de Lyapunov

A estabilidade é de facto uma das características dos sistemas dinâmicos mais importantes, e aquela que nos interessa no âmbito deste trabalho. É sabido que as equações que regem o movimento do ASV, nomeadamente ao nível do *heading*, são equações não lineares. Na medida em que é necessário um controlo do veículo, para que este siga uma determinada trajectória, surgiu a necessidade de estudar a estabilidade do nosso sistema, de modo a que se consiga sintetizar um controlador adequado.

Dada a necessidade de conviver com sistemas não lineares, é necessário recorrer a alternativas que permitam avaliar a estabilidade do sistema dinâmico em questão já que, como é sabido, os métodos usados para sistemas lineares não resultam. Para tentar avaliar então a estabilidade de sistemas não lineares, é usual recorrer ao conceito de estabilidade de Lyapunov.

Segundo o conceito de estabilidade de Lyapunov, é possível determinar a estabilidade de um sistema em relação a um ponto de equilíbrio, e não a estabilidade de um sistema só por si, tal como acontece com sistemas lineares. Isto quer dizer que para um dado ponto de equilíbrio o sistema poderá ser estável, enquanto que para um outro poderá já não o ser. Ou seja, a estabilidade depende não só do sistema em si, mas também do ponto de equilíbrio em questão [12].

Um estado de equilíbrio, ou ponto de equilíbrio  $x_0$ , é um estado particular do sistema, que quando é atingido implica a sua permanência nesse estado. Sempre que um sistema atinge um ponto de equilíbrio, o sistema mantê-se constante, não saindo mais desse estado. Matematicamente, isto representa-se por:

$$0 = f(x_0) \tag{2.14}$$

Um sistema diz-se estável para um determinado ponto de equilíbrio se, quando sujeito a pequenas perturbações, permanece sempre numa região finita que envolve esse mesmo ponto. Dito de outro modo, a estabilidade verifica-se sempre que a trajectória de um sistema se pode manter arbitrariamente próxima do seu estado inicial. Se adicionalmente se verificar não só que a trajectória se mantém arbitrariamente próxima, mas que a trajectória tende para o ponto inicial, então diz-se que esse estado inicial é assintoticamente estável. Adicionalmente, se se verificar que para qualquer estado inicial a trajectória é uma trajectória estável, diz-se que o ponto de equilíbrio atingido é globalmente e assintoticamente estável [13].

### 2.4.2 Funções de Lyapunov

O matemático russo Aleksandr Lyapunov dedicou muito do seu trabalho a estudos no campo das equações diferenciais e estabilidade de sistemas. Com este trabalho desenvolveu a teoria da estabilidade de sistemas, e publicou estudos, hoje em dia conhecidos como os Métodos de Lyapunov, e que permitem uma análise a sistemas não lineares.

O primeiro método de Lyapunov, também conhecido por método de linearização de Lyapunov, baseia-se na linearização de sistemas não lineares de sistemas de equações diferenciais ordinárias, relacionando a estabilidade destes sistemas e os seus pontos de equilíbrio, com a estabilidade observada após linearização.

O segundo método de Lyapunov, ou método directo de Lyapunov, é bastante mais sistemático, permitindo maior liberdade na análise de sistemas não lineares. Baseando-se na observação de sistemas mecânicos, e verificando que se a energia total de tais sistemas se for continuamente dissipando, o sistema tende a estabilizar num ponto de equilíbrio, Lyapunov desenvolveu um método que permite a análise de qualquer sistema não linear.

De acordo com este segundo método, é possível concluir acerca da estabilidade de um determinado sistema, sem que seja necessário obter explicitamente as soluções do sistema, algo por vezes bastante complicado. O método baseia-se em encontrar uma função de Lyapunov para o sistema, analisando depois a sua variação ao longo tempo. Esta variação irá garantir ou não a estabilidade assintótica do sistema, global ou localmente, em torno de um determinado ponto de equilíbrio.

Para que uma função seja considerada função de Lyapunov de um determinado sistema, tem que responder a alguns critérios: em primeiro lugar, a função tem que ser definida ou semi-definida positivamente; já a sua derivada terá que ser definida ou semi-definida negativamente. Se para um dado sistema for possível encontrar uma função que respeite estes critérios localmente, então pode-se concluir acerca da estabilidade local desse mesmo sistema, mas se estes critérios forem respeitados globalmente, então poder-se-á concluir acerca da estabilidade global. Se, pelo contrário, não for possível encontrar uma função deste género, então nada se pode concluir acerca da estabilidade de tal sistema.

Em termos matemáticos, se existir a função  $V(X)$  e se se verificar que:

$$V(X) > 0 \text{ para } X \neq 0 \text{ e } V(X) = 0 \text{ para } X = 0 \quad (2.15)$$

$$\dot{V}(X) < 0 \text{ para } X \neq 0 \text{ e } \dot{V}(X) = 0 \text{ para } X = 0 \quad (2.16)$$

$$V(X) \rightarrow \infty \text{ ||}X\text{||} \rightarrow \infty \quad (2.17)$$

então o equilíbrio na origem é assintoticamente estável.

## 2.5 Síntese

A apresentação destas técnicas, nas secções anteriores, justifica-se dada a sua relevância no contexto do trabalho. Por um lado, o estudo de modelos que reflectam a dinâmica de veículos aquáticos permitiu uma melhor compreensão dos fenómenos físicos envolvidos, bem como da síntese de uma equação que descreva o comportamento desejado do veículo. Por outro lado, estimar a posição do AUV, recorrendo a um filtro de Kalman não seria de todo

possível sem uma correcta compreensão dos métodos de navegação acústica subjacentes. Finalmente, a estabilidade da estratégia de controlo a adoptar será provada recorrendo ao método da estabilidade de Lyapunov.





## Capítulo 3

# Veículos e Estrutura Computacional

Neste capítulo apresentam-se os veículos usados no trabalho e a sua estrutura computacional. Uma vez que é fundamental que se consiga uma integração dos algoritmos desenvolvidos com todo o sistema já existente, nomeadamente com software de bordo do ASV Zarco e com a base de dados (Repositório) onde se registam todos os eventos decorrentes de uma missão, compreender a fundo toda a estrutura que envolve a execução das diferentes missões foi de facto uma exigência decorrente dos objectivos propostos. A integração dos algoritmos desenvolvidos na estrutura computacional já existente está esquematizada na [figure 3.1](#)

Desta forma, nas primeiras duas secções serão apresentados os veículos existentes, o ASV Zarco e o AUV Mares, dando-se uma breve explicação do seu funcionamento. É também referido na secção seguinte o funcionamento dos faróis acústicos e da sua interacção com ambos os veículos. Finalmente é apresentado o modelo de comunicações utilizado no OSG e introduz-se o conceito da entidade Repositório.

### 3.1 AUV MARES

O AUV MARES, acrónimo para Modular Autonomous Robot for Environment Sampling, é um veículo autónomo subaquático, implementado tendo sobretudo em vista a recolha de dados oceanográficos em diferentes ambientes, que podem ir desde estuários de rios e lagos, tendo também já sido testado o seu funcionamento com sucesso em alto mar.

As missões que se pretende que o AUV desempenhe são de natureza bastante diversa. Deste modo, o AUV foi concebido com uma estrutura altamente modular, permitindo alterar a configuração dos diversos sensores facilmente. Para além dos sensores e instrumentação necessária à navegação autónoma, o AUV pode ainda transportar uma variada gama de outros sensores oceanográficos que podem ser facilmente incluídos ou retirados, consoante a missão.

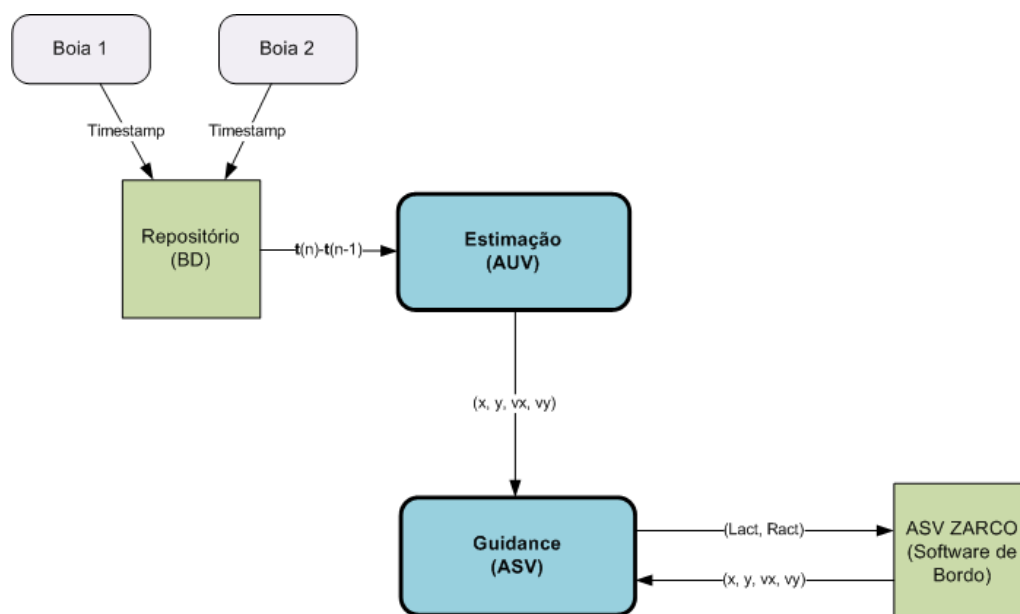


Figura 3.1: Integração dos algoritmos na estrutura computacional existente

As dimensões deste veículo são também um dos seus pontos fortes. O seu comprimento não ultrapassa os 1,5 m e o seu peso é de cerca de 32 kg, podendo ser facilmente transportável.

O MARES é actuado por um conjunto de quatro pequenos motores eléctricos, colocados estrategicamente na sua estrutura de tal forma que o movimento no plano horizontal é praticamente independente do movimento no plano vertical. Estes motores são alimentados por um conjunto de baterias, que fornecem uma autonomia de até 10 horas ou 40 km.

O sistema de navegação deste veículo é feito à custa de *dead-reckoning*, levado a cabo por um conjunto de sensores apropriado, com o auxílio de uma rede acústica LBL. O seu controlo está especificamente indicado para o seguimento de trajectórias rectilíneas, alcançando aí resultados suficientemente satisfatórios para as aplicações a que se destina.



Figura 3.2: AUV MARES

## 3.2 ASV Zarco

O veículo autónomo de superfície, ASV Zarco, assim baptizado em referência ao navegador português João Gonçalves Zarco, é uma embarcação de pequena dimensão, ao estilo *catamaran*, concebido para operações a desempenhar em águas com pouca agitação.

Este veículo, alimentado por um conjunto de baterias que lhe conferem uma autonomia que pode chegar até às 3 horas, é actuado por um conjunto de dois motores colocados na sua popa, atingindo velocidades na ordem dos 2 m/s.

Este veículo pode ser autonomamente controlado, seguindo rotas previamente predefinidas, ou, em alternativa, pode ainda ser controlado remotamente, através de uma ligação WiFi com uma estação terra. Para o controlo autónomo, este veículo está dotado dos sensores de navegação necessários, onde se incluem um receptor GPS de elevada precisão.

Este veículo poderá ainda operar como um farol acústico móvel, já que dispõe de receptores e emissores acústicos subaquáticos, sendo inclusive possível estabelecer comunicação com o AUV MARES.



Figura 3.3: ASV Zarco

## 3.3 Bóias

As bóias, também chamadas de faróis acústicos, são fundamentais na operação dos veículos autónomos subaquáticos, já que por vezes são a única ferramenta de localização de que esses veículos dispõem.

Cada uma das bóias possui, para além de um conjunto de aparelhagem electrónica adicional, um transdutor acústico. Este receptor permite que as bóias detectem todos os sinais acústicos numa gama de frequências alargada e apropriada para a comunicação subaquática. Essas frequências andam, geralmente, na gama dos 20KHz até aos 30 kHz. No entanto, cada uma das bóias encontra-se equipada com um conjunto de filtros programáveis, que permitem a cada uma delas distinguir uma gama de frequências diferentes, frequências às quais deve responder.

Numa missão, o AUV interroga uma das bóias com um sinal acústico numa determinada frequência. Apesar de esse sinal ser recebido por todas as bóias da rede, apenas uma irá responder com um sinal acústico apropriado. Desta forma, é possível que os veículos interroguem cada uma das bóias individualmente, extrapolando daí a sua posição.

Cada sinal acústico enviado ou recebido por cada uma das bóias é denominado evento acústico. Cada evento acústico que ocorra em cada bóias é enviado por uma ligação rádio para um computador central (repositório), juntamente com a identificação da boia que o detectou e ainda com informação relativa ao instante em que aconteceu.

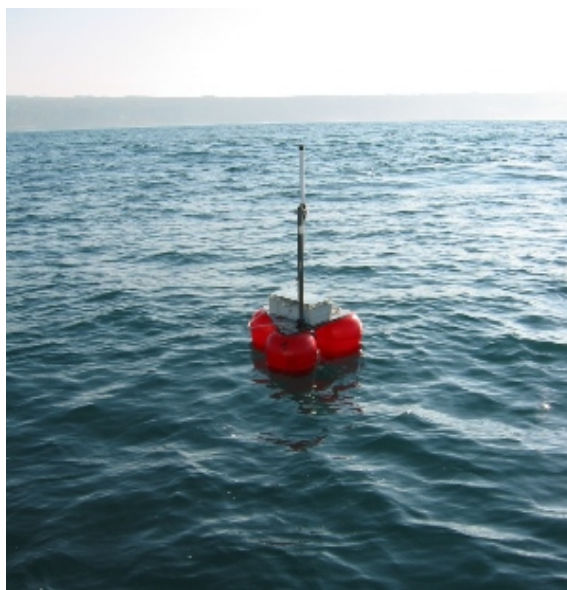


Figura 3.4: Farol Acústico

### 3.4 Modelo de Comunicações

De acordo com [14] o Modelo de Comunicação OceanSys, é estabelecido o conceito de *Entidade* e que se refere a qualquer veículo ou sistema de operação que esteja em missão. Deste modo, cada um dos veículos, ASV ou AUV, em missão é uma entidade distinta, bem como quaisquer bóias que possam integrar a rede acústica.

Cada entidade é identificada individualmente pelo seu endereço e poderá ser constituída por um ou mais subsistemas, estes possuindo cada um o seu endereço individual. Desta forma é possível uma comunicação entre cada subsistema e entre cada entidade.

Exemplificando, o AUV em missão constitui uma Entidade, e tem diferentes subsistemas como o sistema de navegação e controlo, sistema de energia, etc. Desta forma, é possível a qualquer subsistema exterior ao AUV comunicar directamente com um subsistemas em particular.

O Modelo de Comunicação OceanSys estabelece também um modelo de comunicações, utilizando o protocolo IP como base para a troca de mensagens. Em concreto, esta rede

IP é baseada em routers wireless permitindo a comunicação sem fios e à distância entre bóias, ASV e estação base.

As mensagens trocadas entre entidades e subsistemas encontram-se encapsuladas nas mensagens IP e são compostas por 4 campos: 3 de comprimento fixo, e que constituem o cabeçalho da mensagem, e um campo de comprimento variável, e que consiste na própria mensagem a transmitir. Para além do campo *MsgType*, que define qual o tipo de mensagem, logo, o comprimento da mensagem, o cabeçalho de cada mensagem é ainda constituído por campos de *ID* e que identificam a entidade e subsistema de origem ou destino, dependendo do tipo de mensagem.

| Campo   | Descrição                 |
|---------|---------------------------|
| Id      | Id da entidade destino    |
| SubId   | SubId da entidade destino |
| MsgType | Tipo da Mensagem          |
| Payload | Dados da Mensagem         |

Tabela 3.1: Estrutura das Mensagens Trocadas

## 3.5 Repositório

Para suporte às missões, é ainda utilizado ainda um repositório de informações, para onde são canalizados em tempo real todos os dados considerados relevantes. Este repositório permite não só, e de um modo fácil, centralizar todas as informações e comunicações do sistema, permitindo ainda o seu armazenamento, facto muito importante para detectar e resolver problemas que possam eventualmente surgir no decorrer de cada missão.

Todos os dados recebidos pelo repositório poderão ainda ser reencaminhados, consoante tenham ou não interesse para as restantes entidades e/ou subsistemas. No caso concreto deste trabalho, os algoritmos desenvolvidos necessitam de alguns dados provenientes de outros subsistemas, nomeadamente nas bóias, dados esses que serão enviados pelo repositório.

De referir ainda que inerentes ao processo de transmissão dos dados dos vários subsistemas e posterior armazenamento destes no repositório existem e são bastante significativos, já que podem chegar às centenas de milissegundos. Estes atrasos, variáveis e difíceis de prever, têm contudo pouca importância relativa. Considerando as velocidades de ambos os veículos em missão, tipicamente na ordem dos 1 m/s a 2 m/s, os erros decorrentes destes atrasos na estimação da posição actual dos veículos são, de facto, muito residuais.



## Capítulo 4

# Algoritmos Desenvolvidos

Este capítulo encontra-se dividido em duas partes distintas e aborda os diferentes algoritmos desenvolvidos, necessários para que se atinjam os objectivos propostos.

Na primeira secção é apresentado o problema da estimação da posição do AUV, fundamental para que os resultados no seu global sejam satisfatórios. São ainda discutidas as várias particularidades inerentes a este processo, bem como soluções que o permitam otimizar. Na segunda parte, é abordado o algoritmo de *guidance*, ou seja o algoritmo responsável por actuar o ASV de forma mais indicada para que as trajectórias deste e do AUV coincidam.

De salientar que pormenores mais específicos de implementação, e que se prendam com aspectos mais técnicos de programação, mas que mereçam algum destaque dadas as suas peculiaridades serão apresentados em anexo. Da mesma forma, a discussão relativa a estratégias de controlo a implementar será apresentada no capítulo seguinte.

### 4.1 Estimação da Posição do AUV

Para que as trajectórias do AUV e ASV coincidam, objectivo final do trabalho, é necessário em primeiro lugar estimar em tempo real a posição do AUV. Nesse sentido, foi desenvolvido um algoritmo que o permite, aproveitando as particularidades da navegação apoiada em redes acústicas de posicionamento.

Para a operação dos seus veículos autónomos, o OSG tem disponível uma rede acústica composta por vários faróis acústicos, que auxilia nas tarefas de localização dos veículos. Contudo, e sempre que as missões o permitem, aproveita-se o facto de ser possível operar apenas com duas bóias por razões de simplicidade das operações.

No caso específico dos veículos subaquáticos, e em concreto do AUV MARES, este possui um sistema electrónico de *dead reckoning*, que permite a sua localização relativa no espaço mas apenas durante curtos períodos de tempo, após os quais os dados por aí obtidos ficam seriamente comprometidos por ruído. Para ultrapassar estas limitações, o

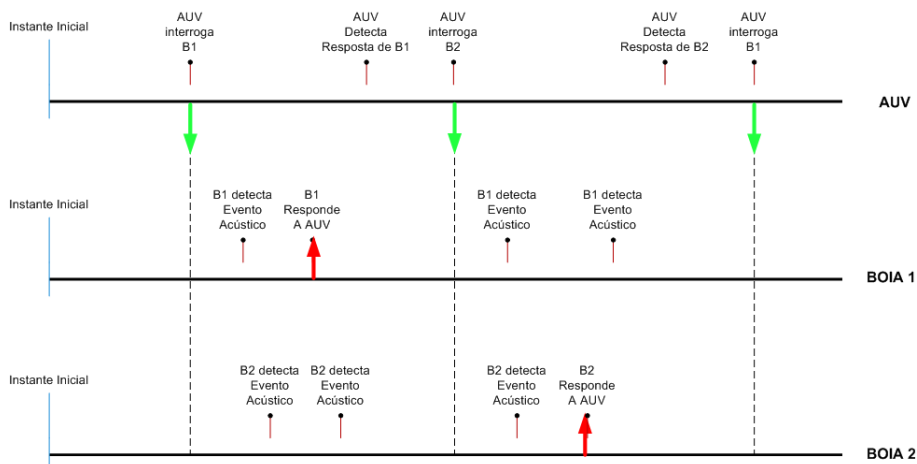


Figura 4.1: Troca de sinais acústicos: esquema

AUV está dotado de um transdutor acústico, que lhe permite trocar informações com a referida rede, permitindo a sua localização num referencial previamente estabelecido.

#### 4.1.1 Algoritmo de Estimação

No funcionamento de uma rede acústica, os veículos devem interrogar cada uma das bóias que a compõem, tal como descrito no capítulo 2, sendo possível através da medição do tempo de resposta extrapolar a distância entre o veículo e cada uma das bóias. Conhecendo este processo, e tendo acesso a estas medidas, é então possível para uma entidade exterior estimar a posição do AUV, baseada no registo das comunicações entre todas as entidades.

Com base num algoritmo proposto por [15], e conhecendo a forma como todo o processo de comunicações, neste caso concreto, se desenrola, foi possível estabelecer um modelo das comunicações efectuadas ciclicamente. Este modelo de comunicações está esquematizado na figura 4.1 e apresentado em detalhe na tabela 4.1.

Tal como esquematizado na figura 4.2, o AUV encontra-se à distância  $d_1$  da boia 1 e à distância  $d_2$  da boia 2. Por outro lado, um sinal acústico demora  $t_1$  segundos a ir da posição do AUV à boia 1 e  $t_2$  segundos a ir da posição do AUV à boia 2, considerando que a velocidade de propagação dos sinais acústicos na água é constante e igual a  $v_{som}$ . De facto, e apesar de a velocidade de propagação de sinais acústicos ser variável com parâmetros tão distintos como temperatura, salinidade, pressão ou ainda profundidade, é possível considerar, por simplicidade, que  $v_{som} = 1506m/s$  em qualquer situação, sem que sejam cometidos erros de aproximação significativos.

$$\frac{d_1}{t_1} = \frac{d_2}{t_2} = v_{som} \quad (4.1)$$

Em cada ciclo de interrogações, durante o qual se admite que a posição do AUV se mantém inalterada, este envia a cada uma das bóias um pedido, na forma de sinal acústico,



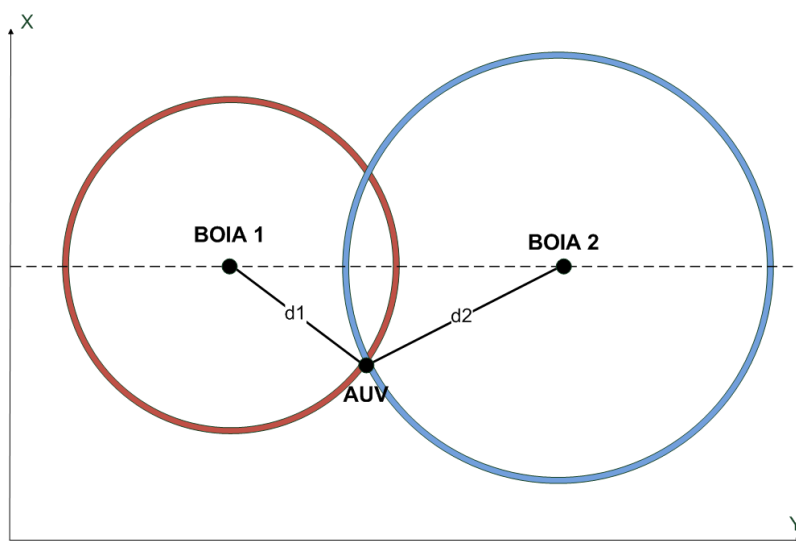


Figura 4.2: Esquemática de uma situação de missão típica

esperando depois pelas suas respostas. Na tabela 4.1 é então possível ver detalhadamente cada um destes ciclos de comunicações. O instante em que se admite que se iniciam as comunicações é denominado por  $t_0$  e  $\delta_X$  refere-se ao atraso entre a recepção de um evento acústico de uma boia, e a resposta por ela emitida. Já  $\delta_Y$  representa o tempo decorrido entre a recepção de um sinal de resposta por parte do AUV, e a subsequente interrogação a outra boia. Admite-se ainda, por questões de simplicidade, que AUV se encontra mais próximo da boia 1 do que da boia 2, não tendo este facto influência no desempenho do algoritmo.

Tabela 4.1: Ciclo de Comunicações

|          | Instante  | Descrição                 |
|----------|---|---------------------------|
|          | $t_0$   | AUV interroga B1          |
| $t_1$    | $t_0 + \Delta t_1$  | B1 recebe pedido de AUV   |
| $t_2$    | $t_0 + \Delta t_2$  | B2 recebe pedido de AUV   |
| $t_3$    | $t_0 + \Delta t_1 + \delta_X$   | B1 responde a AUV         |
| $t_4$    | $t_0 + \Delta t_1 + \delta_X + \Delta t_{12}$                           | B2 recebe resposta de B1  |
| $t_5$    | $t_0 + 2\Delta t_1 + \delta_X$  | AUV recebe resposta de B1 |
| $t_6$    | $t_0 + 2\Delta t_1 + \delta_X + \delta_Y$                               | AUV interroga B2          |
| $t_7$    | $t_0 + 2\Delta t_1 + \Delta t_2 + \delta_X + \delta_Y$                  | B2 recebe pedido de AUV   |
| $t_8$    | $t_0 + 3\Delta t_1 + \delta_X + \delta_Y$                               | B1 recebe pedido de AUV   |
| $t_9$    | $t_0 + 2\Delta t_1 + \Delta t_2 + 2\delta_X + \delta_Y$                 | B2 responde a AUV         |
| $t_{10}$ | $t_0 + 2\Delta t_1 + \Delta t_2 + 2\delta_X + \delta_Y + \Delta t_{12}$ | B1 recebe resposta de B2  |
| $t_{11}$ | $t_0 + 2\Delta t_1 + 2\Delta t_2 + 2\delta_X + \delta_Y$                | B2 responde a AUV         |
| $t_{12}$ | $t_0 + 2\Delta t_1 + 2\Delta t_2 + 2\delta_X + 2\delta_Y$               | AUV interroga B1          |

Tendo em atenção esta troca de mensagens, é possível estabelecermos uma relação entre

os instantes em que cada um dos eventos acústicos acontecem. De facto, os instantes de tempo em que cada uma das bóias escuta os pedidos do AUV são suficientes para estimar a posição do AUV.

O AUV emite sinais acústicos, interrogando alternadamente as bóias 1 (B1) e 2 (B2). Considerando que o ciclo se inicia com uma interrogação a B1, chega-se à tabela 4.1. Atentando em todos os instantes de tempo a que é possível aceder instantaneamente, isto é, a todos os eventos acústicos detectados por cada um dos faróis acústicos, é possível calcular as distâncias  $d_1$  e  $d_2$ : se subtrairmos o instante de tempo  $t_8$ , em que B1 regista a recepção do sinal destinado a B2, ao instante  $t_1$ , em que B1 regista a recepção do sinal a B1 obtem-se o seguinte:

$$t_8 - t_1 = 2\Delta t_1 + \delta_X + \delta_Y \quad (4.2)$$

Os intervalos de tempo  $\delta_X$  e  $\delta_Y$  são conhecidos, sendo uma imposição de design do próprio sistema.  $\delta_X$ , correspondente a um atraso da resposta bóia à sua interrogação, depende apenas de questões de Hardware, sendo portanto um valor bastante baixo e determinístico. Por outro lado  $\delta_Y$  depende exclusivamente do sistema computacional e é um atraso implementado por Software, estando sujeito portanto a um *jitter mais elevado*; em concreto, pode-se afirmar que variações na ordem de 1 ms darão origem a variações de 1 a 2 metros no cálculo da posição. Sendo  $v_{som}$  a velocidade de propagação das ondas acústicas na água, é então possível calcular a distância do AUV a B1, já que esta é proporcional a (4.2).

$$d_1 = \frac{t_8 - t_1 - \delta_X - \delta_Y}{2v_{som}} \quad (4.3)$$

Por outro lado, também se pode notar, mais uma vez recorrendo à tabela 4.1, que B2 também detecta a posição do AUV em relação a B1, ou seja, também é possível calcular  $d_1$  partindo dos dados de B2. De facto, se subtrairmos  $t_7$  a  $t_2$  obtém-se o mesmo que em (4.2), podendo, de forma análoga a (4.3) obter  $d_1$ . Para calcular  $d_2$  utiliza-se o mesmo processo, começando cada ciclo com a interrogação de B2.

Sintetizando, com os dados fornecidos por apenas uma das bóias, é possível calcular a distância que separa o AUV a cada uma delas. Subtraindo os instantes de tempo em que B2 é interrogado aos instantes de tempo em que B1 é interrogado é possível calcular a distância do AUV à boia 1. Analogamente, subtraindo os instantes de tempo em que B1 é interrogado aos instantes de tempo em que B2 é interrogado é possível calcular a distância do AUV à boia 2. Este comportamento, pode ser esquematizado tal como a máquina de estados da figura 4.3.

#### 4.1.2 filtro de Kalman

Na secção anterior foi descrito o processo de obter a distância do AUV à cada uma das bóias da rede acústica. No entanto, o objectivo final é estimar a dinâmica do AUV, ou seja

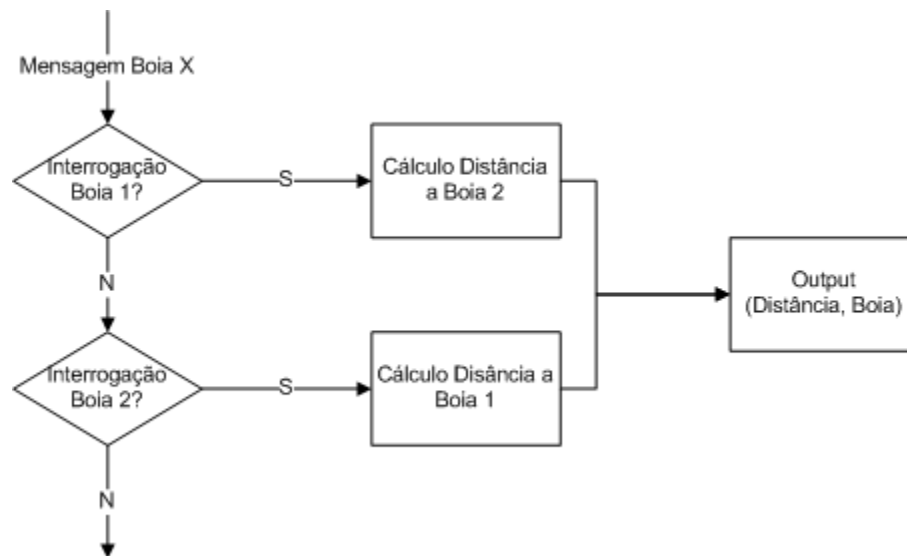


Figura 4.3: Fluxograma das máquinas de estados de cada uma das bóias

a sua posição e as suas velocidades no referencial definido pelas duas bóias, pois só assim será possível que as duas trajectórias sejam semelhantes.

Interrogando a rede acústica consegue-se determinar a posição do AUV no referencial definido pelas bóias. No entanto, este método de actualização é um processo discreto e com taxas de actualização na ordem dos 1 Hz, ou até inferior. Por outro lado, para além de, idealmente, ser necessário estabelecer continuamente a posição do AUV para determinar correctamente a sua trajectória, também se verifica que a actuação do ASV necessita de comandos apropriados a uma frequência de aproximadamente 50 Hz.

Usando um filtro de Kalman contínuo-discreto é possível, com um modelo apropriado do sistema, prever a evolução do sistema, corrigindo estas evoluções sempre que se dispõe de novas medidas de distância.

Tal como descrito em [16], e considerando que durante as missões o AUV se move fundamentalmente ao longo de linhas rectas, é admissível modelar a dinâmica do AUV do seguinte modo:

$$\begin{bmatrix} \dot{x} \\ \dot{v}_x \\ \dot{y} \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \beta \end{bmatrix} \begin{bmatrix} x \\ v_x \\ y \\ v_y \end{bmatrix} + \begin{bmatrix} n_x \\ n_{vx} \\ n_y \\ n_{vy} \end{bmatrix} \quad (4.4)$$

Em (4.4)  $(x,y)$  refere-se às coordenadas do veículo no plano horizontal, enquanto  $(v_x, v_y)$  refere-se às velocidades em relação a um referencial fixo. O parâmetro  $\beta$  representa um decaimento exponencial da velocidade, e encontra justificação na necessidade que as duas componentes da velocidade tendam para zero se, por qualquer motivo, não chegarem novas medidas. Desta forma garante-se que, no limite, o nosso AUV irá parar. O vector

$(n_x, n_{vx}, n_y, n_x)^T$  modeliza as diferenças entre o movimento real do veículo e o movimento previsto por este modelo.

Usado um filtro contínuo-discreto de Kalman é então possível integrar as medidas de distância obtidas através da rede acústica com uma previsão do comportamento do sistema.

Sempre que se dispõe de uma nova medida, actualizam-se as matrizes de estado e covariância,  $X_k$  e  $P^k$ , para novos valores.

$$X_k^+ = X_k^- + K_k(z_k - z_k^*) \quad (4.5)$$

$$P_k^+ = (I - K_k H) P_k^- \quad (4.6)$$

Nestas equações,  $z_k$  identifica a nova medida de distância, e  $z_k^*$  identifica a previsão dada pelo filtro de Kalman para essa mesma distancia. Para uma boia posicionada nas coordenadas  $(x_0, y_0)$ ,  $z_k^*$  é calculado com base na previsão do estado do sistema, tal como em (4.7).

$$z_k^* = \sqrt{(x - x_0)^2 - (y - y_0)^2} \quad (4.7)$$

A matriz  $H$  relaciona as medidas efectuadas com o estado do sistema e, estando nós na presença de um EKF contínuo-discreto, é dada pela matriz das derivadas parciais.

$$H = \frac{\partial z_k}{\partial X} = \begin{bmatrix} \frac{x-x_0}{\sqrt{(x-x_0)^2+(y-y_0)^2}} & 0 & \frac{y-y_0}{\sqrt{(x-x_0)^2+(y-y_0)^2}} & 0 \end{bmatrix} \quad (4.8)$$

A matriz de ganho  $K_k$  é calculada tal como indicado em (4.9) e  $r_s$  representa a covariância do erro associado à medida de  $z_k$ .

$$K_k = P_k^- H^T S_k^{-1} \quad (4.9)$$

$$S_k = H P_k^- H^T + r_s \quad (4.10)$$

Entre cada duas medidas consecutivas, que permitam obter a distância entre o AUV e as bóias, é necessário pois prever a evolução do sistema. Partindo do modelo do sistema (4.4), actualiza-se a evolução do sistema com base nas equações (4.11) e (4.12), que prevêm a evolução contínua do estado do sistema e da sua matriz de covariância, considerando que  $t = t_{k+1} - t_k$  [15].

$$X_{k+1} = e^{At} X_k^+ \quad (4.11)$$

$$P_{k+1} = e^{At} (P_k + \int_0^t e^{-As} Q e^{-A^T s} ds) e^{A^T t} \quad (4.12)$$

Este modelo de filtro de Kalman Estendido tem pois como grande vantagem a fácil linearização das medidas obtidas, à custa da utilização das derivadas parciais, tal como descrito no capítulo 2. Para além disso, este filtro permite ainda fazer uma avaliação em relação à qualidade das medidas, permitindo desprezar medidas que não respeitem (4.13)

$$\|z_k - z_k^*\|_{S_k^{-1}} \leq \gamma \quad (4.13)$$

$\gamma$  traduz a confiança que se tem na estimação e depende, obviamente, na qualidade das medidas obtidas, medidas que actualizam o filtro de Kalman, como discutido ao longo desta secção.

Neste caso em concreto, em que as medidas traduzem o acontecimento de eventos acústicos,  $2 < \gamma < 4$  para situações em que a interferência existente nos sinais acústicos trocáveis é bastante baixa. Por outro lado, sempre que as condições do meio não o permitem, este parâmetro poderá subir para valores bem mais altos, como  $20 < \gamma < 30$ . A quantificação destes valores de  $\gamma$  pode ser feita, tendo em conta o *know-how* e a experiência adquirida ao longo do tempo pelo OSG, decorrente das várias missões já levadas a cabo.

Apesar disso, a estimação da dinâmica não é completamente satisfatória. De facto, apesar de as estimativas da posição  $(x, y)$  serem bastante fiáveis, o mesmo não acontece para as estimativas da velocidade  $(v_x, v_y)$ . O mesmo quer dizer que, como o modelo (4.4) não fornece grande informação em relação ao comportamento da velocidade para além de prever o seu decaimento exponencial, as várias estimativas produzidas contém pouca informação relevante. Para obter estimativas mais animadoras, é preciso também acrescentar alguma informação aos modelos, usando-se para isso a informação que as trajectórias efectuadas pelo AUV são rectilíneas.

### 4.1.3 Estimador Recursivo de Mínimos Quadrados

Tal como já foi dito anteriormente, assume-se que grande parte do movimento do AUV é feito em linhas rectas e a velocidade constante. Assim, as equações que regem o movimento do AUV, para cada recta, podem ser descritas tal como em (4.14) e (4.15).

$$x(t) = x_0 + v_x t \quad (4.14)$$

$$y(t) = y_0 + v_y t \quad (4.15)$$

Nestas equações está descrito o movimento do AUV está desacoplado segundo as suas componentes  $x$  e  $y$ , respectivamente.  $(x_0, y_0)$  são as posições no plano horizontal para o instante  $t = 0$  e  $(v_x, v_y)$  as velocidades, também no plano horizontal.

Estas equações são lineares e podem pois ser utilizadas para, com o auxílio de um algoritmo de mínimos quadrados com memória exponencial, ou mínimos quadrados com

esquecimento, obter estimativas bastante melhores que as dadas pelo filtro de Kalman para as velocidades do AUV no plano horizontal [16].

O algoritmo dos mínimos quadrados recursivo passa pelo cálculo da matriz de ganho, dos parâmetros e da covariância do erro, de forma indicada em (4.16), (4.17) e (4.18), [17].

$$K_n = \frac{P_{n-1}\phi_n}{\lambda + \phi_n^T P_{n-1} \phi_n} \quad (4.16)$$

$$\theta_n = \theta_{n-1} + K_n[y_n - \phi_n^T \theta_{n-1}] \quad (4.17)$$

$$P_n = [I - K_n \phi_n^T] P_{n-1} \frac{1}{\lambda} \quad (4.18)$$

Adaptando estas equações ao nosso caso concreto, os parâmetros variantes que se quer estimar são as posições  $(x_0, y_0)$  e as velocidades  $(v_x, v_y)$ , a partir dos dados que dispomos, isto é  $(x, y)$  e o instante em que ocorrem,  $t_i$ . Exemplificando para o caso relativo às componentes do movimento segundo  $x$ , e tomando em consideração (4.14), ficaria:

$$\theta = \begin{bmatrix} x(t) \end{bmatrix} \quad (4.19)$$

$$\phi^T = \begin{bmatrix} 1 & t_i \end{bmatrix} \quad (4.20)$$

$$y = \begin{bmatrix} x_0 \\ v_x \end{bmatrix} \quad (4.21)$$

O algoritmo de mínimos quadrados recursivo com esquecimento é largamente usado em aplicações nas áreas de engenharia, e tem ótimos resultados sempre que a excitação do sistema é suficientemente alta. Quando isto não acontece, há a possibilidade de ocorrerem fenómenos de *wind up* que conduzem ao crescimento exponencial da matriz de covariância, tornando o estimador demasiado sensível e susceptível a erros significativos de aproximação [18].

A diferença fundamental entre o algoritmo dos mínimos quadrados original para o algoritmo dos mínimos quadrados com esquecimento prende-se com a existência de  $\lambda$  em (4.16). É a existência deste factor de esquecimento que permite a estimação de parâmetros variantes no tempo. O seu valor deverá oscilar entre 0 e 1.

O conceito de esquecimento implica que, à medida que o tempo passa, os dados mais antigos são, gradualmente ignorados, à medida que nova informação é incorporada no sistema. Dito de outra forma, o conceito de esquecimento pode ser entendido como a atribuição de um peso superior a dados mais recentes, e a atribuição de um peso inferior para dados mais antigos.

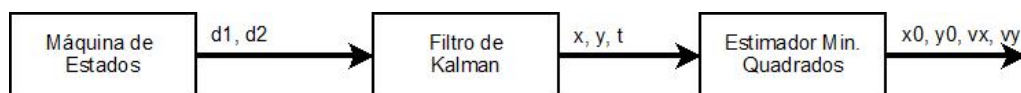


Figura 4.4: Diagrama de blocos para a estimação da posição do AUV

A quantidade de dados, ou histórico, com influência na obtenção de novas estimativas para os parâmetros a determinar pode ser relacionado com o valor de  $\lambda$ . Considerando que, indirectamente, o factor de esquecimento determina a quantidade de tempo que um determinado conjunto de dados tem influência nas estimativas futuras, é então possível aproximar  $\lambda$  por um atraso (4.22). Sendo  $\Delta t$  o intervalo de tempo entre cada nova medida, na ordem dos 2 a 3 segundos, e  $\tau$  é a constante de tempo do sistema, ou seja, o tempo de atraso desejado.

$$\lambda \approx e^{-\frac{\Delta t}{\tau}} \quad (4.22)$$

Sintetizando, a solução proposta para a estimação do AUV está ilustrada na figura 4.4. Inicialmente temos uma máquina de estados que, com base nos intervalos de tempo entre cada interrogação do AUV às várias bóias, calcula a distância entre eles e a posição relativa do AUV num referencial constituído pelas bóias. De seguida, usando um filtro de Kalman, é possível obter boas estimativas da posição do AUV, o mesmo não acontecendo com as velocidades. O filtro de Kalman permite ainda linearizar as medidas obtidas bem como fazer a sua validação eliminando medidas erróneas, já que analisando a matriz das covariâncias consegue-se avaliar o grau de fiabilidade de cada medida. Finalmente, utilizando um algoritmo de mínimos quadrados que modeliza o movimento do AUV em linhas rectas, conseguimos obter boas estimativas da dinâmica global do sistema, isto é, estimativas para a posição  $(x, y)$  e para a velocidade  $(v_x, v_y)$ .

## 4.2 Guidance do ASV

O algoritmo de controlo do ASV, ou algoritmo de *guidance*, tem um funcionamento muito simples e o seu objectivo é calcular as variáveis de controlo e actuação necessárias para que o ASV tenha a trajectória pretendida.

Para isso, este algoritmo necessita de saber não só as posições e velocidades que caracterizam a dinâmica do AUV, bem como as grandezas que caracterizam o movimento do AUV.

Na prática isto significa que este algoritmo se pode dividir em duas partes distintas, que funcionam simultaneamente. Uma parte está encarregue de receber as mensagens relevantes, com as informações respeitantes ao AUV e ASV. A outra parte está encarregue de, com os dados recebidos, implementar as estratégias de controlo mais adequadas. Uma discussão das estratégias de controlo será feita no capítulo seguinte.

O *software* de bordo do ASV aceita como comando para cada um dos motores valores entre -100 e 100. Ou seja, cada motor é controlado em percentagem da força máxima que consegue desenvolver, quer no sentido positivo, quer no sentido negativo.

Sabe-se que, para o ASV Zarco, a força máxima de cada um dos motores é de 130N. No entanto, esta informação não é suficiente, já que os controladores a desenvolver, a abordar no capítulo seguinte, conseguem obter valores tanto para a força total  $X$ , como para o binário,  $N$ , desenvolvido em relação aos respectivos eixos. Desta forma, vem:

$$\begin{aligned} X_{act} &= F_1 + F_2 \\ N_{act} &= (F_1 - F_2)\frac{b}{2} \end{aligned} \quad (4.23)$$

Em (4.23)  $F_1$  e  $F_2$  são a força que cada um dos motores provoca e  $b$  é o braço entre elas, ou seja, a distância entre os motores. Este valor depende da configuração dos motores e no ASV Zarco actualmente é de 48.6cm.

Já que o com o controlo do veículo, tanto em velocidade como em direcção, iremos conseguir valores para  $X_{act}$  e  $N_{act}$ , então vem:

$$\begin{aligned} F_1 &= \frac{X_{act}}{2} + \frac{N_{act}}{b} \\ F_2 &= \frac{X_{act}}{2} - \frac{N_{act}}{b} \end{aligned} \quad (4.24)$$



# Capítulo 5

## Controlo do ASV

Tal como descrito no capítulo 2, partindo de um modelo geral que descreve o movimento de um corpo rígido marinho (2.1), é possível chegar a um modelo simplificado do movimento de um barco, modelo esse aplicável ao nosso ASV e com apenas três graus de liberdade. Em (2.3), (2.4) e (2.5) podemos ver as três equações desse modelo, sendo a primeira a equação de velocidade do veículo. Por seu lado, as restantes equações definem a direcção do veículo.

Este capítulo é dedicado ao controlo do ASV. Sendo possível desacoplar o movimento do veículo nas suas componentes, inicialmente irá ser abordado o controlo de direcção, ou controlo de *heading* do veículo, e na secção seguinte discute-se o seu controlo de velocidade.

### 5.1 Controlo de Heading

O controlo de *Heading*, ou controlo de direcção, prende-se com a orientação que o nosso ASV deverá ter no plano horizontal de tal forma que se dirija para o AUV.

Considerando de novo o modelo simplificado com 3 graus de liberdade para a dinâmica do veículo, as equações respeitantes à direcção do ASV podem ser reescritas tal como em (5.1) e (5.2), em que  $u_0$  é o módulo de velocidade de cruzeiro pretendido para o ASV.

$$Y = m(\dot{v} - u_0 r) \tag{5.1}$$

$$N = I_z \dot{r} \tag{5.2}$$

Relembremos também que se pretende que o nosso ASV siga, fundamentalmente trajectórias rectilíneas. Ao fazermos esta assunção, podemos de algum modo simplificar estas equações, já que se assume desde logo que  $v$  e  $r$ , respectivamente a velocidade transversal, e a rotação em torno do eixo vertical, são pequenas. Em contrapartida, vai-se traduzir, em princípio, numa quebra de desempenho sempre que o AUV efectuar uma mudança de direcção brusca.

Em [1] são sugeridas várias estratégias e abordagens de controlo, não só em termos de *heading*, como também de *course keeping* e *tracking*. Considerando as estratégias aí abordadas, e não esquecendo as particularidades das trajectórias que se pretende seguir, foi possível escrever (5.3), a lei de controlo para o nosso veículo [16].  $u_0$  é a velocidade de cruzeiro do barco e  $c_y$  é a corrente da água, difícil de determinar e, em geral, desconhecida. É importante ainda referir que  $|c_y| \leq u_0$ , significando isto que a actuação do ASV é suficientemente potente para superar a forças das correntes marinhas e, portanto, controlar eficazmente o ASV.

$$\dot{y} = u_0 \sin(\varphi) + c_y \quad (5.3)$$

De facto, (5.3) representa a dinâmica para o seguimento da linha definida pelo eixo  $x$  do referencial, mas é possível mostrar que esta lei se aplica no caso geral do seguimento de uma linha [16]. Neste caso,  $\varphi$  é a nossa variável de controlo, ou seja, aquilo que é passível de ser manipulável para conseguir um controlo adequado.

A ideia subjacente a este controlador é de facto muito simples e pretende implementar o seguinte comportamento: quando o ASV se encontra bastante longe do AUV, o primeiro deve ir à velocidade de cruzeiro e numa direcção perpendicular à trajectória seguida pelo AUV. O facto de as trajectórias entre ambos os veículos serem perpendiculares, garante-nos que o ASV irá de encontro à trajectória do AUV de acordo com o caminho mais curto.

Por outro lado, à medida que os veículos começam a ficar cada vez mais próximos, o ASV deve ajustar a sua velocidade e o seu heading, de tal forma que as trajectórias de ambos se aproximem. Ou seja, se o ASV estiver suficientemente próximo do AUV, deve então deixar a trajectória perpendicular que mantinha, suavizando a sua aproximação ao ASV. Pretende-se que o comportamento seja algo semelhante com a figura (5.1) em que se pode ver, em linha tracejada, o comportamento desejado para o seguimento de uma trajectória rectilínea coincidente com o eixo  $xx$ . É com base nesta filosofia que  $\varphi$  deve ser concebido.

Considerando a natureza não linear de (5.3), seria também interessante escolher  $\varphi$  de tal forma se procurasse contrariar a não linearidade presente, na forma de uma função sinusoidal. Com isto, facilita-se significativamente a análise de estabilidade do controlo, discutida mais à frente.

$$\varphi = -\arcsin(k_1 y) \quad (5.4)$$

A equação em (5.4) traduz de facto o comportamento desejado, e é válida enquanto  $|k_1 y| \leq 1$ .  $k_1$  é uma constante, e determina a distância entre os dois veículos a partir da qual o ASV deve ajustar a sua velocidade. Por outro lado,  $y$  é a distância de *cross track error* tal como indicado na figura (5.1).

Considerando os parâmetros da trajectória seguida pelo AUV,  $x_0, y_0, v_x$  e  $v_y$ , estimados pelo algoritmo de mínimos quadrados, e considerando ainda a posição actual do ASV, dada

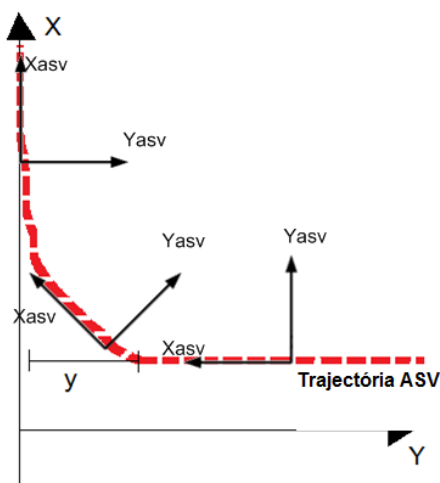


Figura 5.1: Exemplo de comportamento desejado para o ASV

por  $x$  e  $y$ , é possível calcular a distância de *cross track error*, aqui chamada de  $d$ , para não provocar confusão com a posição do ASV [16]. Sendo  $v_{AUV}$  o módulo da velocidade do AUV, dado por  $v_{AUV} = \sqrt{v_x^2 + v_y^2}$ , vem:

$$d = \frac{v_x y_{ASV} - v_y x_{ASV}}{v_{AUV}} + h_{AUV} \quad (5.5)$$

Por seu lado  $h_{AUV}$  representa a diferença entre a posição actual do AUV e a origem da trajectória rectilínea que se presume que está a seguir, trajectória essa estimada através dos parâmetros estimados pelo mínimos quadrados.

$$h_{AUV} = \frac{v_y x_0 - v_x y_0}{v_{AUV}} \quad (5.6)$$

A existência das correntes marinhas, representadas na lei de controlo por  $c_y$ , é também bastante importante. De facto, a imprevisibilidade destas correntes, que dependem de vários factores, como por exemplo as marés, os ventos ou até o local em questão, faz com que seja virtualmente impossível traduzir em números este fenómeno para a generalidade das situações. Por outro lado, o processo de determinar em tempo real a velocidade das correntes é também um processo dispendioso, não produzindo resultados precisos e consistentes. Desta forma, o valor das correntes, em cada dado instante é uma incerteza com que é necessário lidar.

Esta componente irá funcionar então como uma perturbação ao sistema, podendo ser representado como um sinal do tipo degrau, com magnitude igual ao valor absoluto da corrente. De acordo com as teorias de controlo linear, para que um sistema sujeito a uma perturbação do tipo degrau siga com erro nulo a entrada, terá que ser forçosamente um sistema do tipo 1 ou superior. Na prática, isto significa que o sistema precisa de ter, no mínimo um polo na origem, ou, por outras palavras, o sistema necessita de um integrador puro [19].

Tal como descrito em (5.3), o nosso sistema não possui nenhum integrador. Na prática, isto traduz-se na impossibilidade de o ASV seguir a trajectória do ASV a menos de um erro sistemático, erro esse dependente do valor de  $c_y$ .

Para eliminar um erro de resposta ao degrau em regime permanente, é necessário usar um integrador. Para isso, é necessário considerar um estado adicional (5.8), cujo comportamento permita anular o erro em regime permanente (5.7).

$$\varphi = -\arcsin(k_1y + k_2l) \quad (5.7)$$

$$\dot{l} = y \quad (5.8)$$

Considerando o domínio da função  $\arcsin$ , então terá que ser respeitada a condição  $|k_1y + k_2l| < 1$ . Contudo, é desejável um comportamento suave da acção de controlo, não sendo desejável a existência de descontinuidades. Dessa forma, é com naturalidade que (5.7) é definida em todo o seu domínio da seguinte forma, onde  $k_1$  e  $k_2$  são constantes positivas.

$$\varphi = \begin{cases} -\arcsin(k_1y + k_2l) & \text{se } |k_1y + k_2l| < 1 \\ \frac{\pi}{2} & \text{se } k_1y + k_2l < -1 \\ -\frac{\pi}{2} & \text{se } k_1y + k_2l > 1 \end{cases} \quad (5.9)$$

O estado adicional  $l$ , foi introduzido com o intuito de superar erros de regime permanente provocados pela existência de correntes marítimas,  $c_y$ , difíceis de determinar. Na prática, o que este estado faz é integrar a diferença entre as trajectórias executadas por ambos os veículos.

Em relação a este tipo de controlo, integrador, é necessário ter algumas precauções relativamente à sua implementação. De facto, quando se trata de integradores, é preciso garantir que não ocorre a saturação destes, ou seja, é obrigatório salvaguardar que não ocorram fenómenos de *wind-up*. Para isso, a dinâmica do estado adicional  $l$  pode ser definido tal como em (5.10).

$$\dot{l} = \begin{cases} 0 & \text{se } l > \frac{1}{k_2} \wedge y > 0 \\ 0 & \text{se } l < -\frac{1}{k_2} \wedge y < 0 \\ y & \text{restantes} \end{cases} \quad (5.10)$$

### 5.1.1 Estabilidade do Controlo

É naturalmente importante determinar a estabilidade de (5.3), com  $\varphi$  dada por (5.9) e (5.10), sendo para isso necessário determinar uma função de Lyapunov apropriada, já que se está perante uma função não linear. De acordo com [12] para  $V(x)$  ser uma função linear terá que ser semi-definida positivamente, e a sua derivada terá que ser semi-definida negativamente, tal como descrito pelas equações (2.15) a (2.17).

Tendo então como objectivo que a função  $V(x)$  seja sempre positiva e não limitada, será portanto adequado se  $V(x)$  for uma função quadrática de  $y$ , garantindo-se assim que  $V(x) > 0$  no seu domínio. Por outro lado, isto também permite que em  $\dot{V}(x)$  apareçam não só os termos relativos a  $y$  mas também termos relativos a  $\dot{y}$ . A existência destes dois factores será bastante importante para tentar provar que  $\dot{V}(x)$  é semi-definida negativamente.

Após considerar todas estas questões, foi possível chegar a (5.11) uma função de Lyapunov e, por conseguinte, a (5.12). Estas equações permitem provar que (5.3), (5.9) e (5.10) representam um sistema de equações globalmente assintoticamente estável.

$$V(x) = \frac{1}{2} [y^2 + (k_2 l u - c_y)^2] \quad (5.11)$$

$$\dot{V}(x) = y\dot{y} + (k_2 l u - c_y)\dot{l} \quad (5.12)$$

Provar que (5.11) é positivamente semi-definida é trivial, já que estamos perante a presença de uma soma de quadrados. Desta forma,  $V(x)$  será sempre positivo. Por outro lado,  $V(x)$  é também não limitada, dependendo de  $y$  e sendo tão grande quanto se queira.

Garantir agora que  $\dot{V}(x) \leq 0$  é mais trabalhoso, já que implica considerar cada um dos 3 casos que definem tanto (5.9) como (5.10). No total, será portanto necessário verificar o comportamento desta função para o total dos 9 casos, conjugando os 3 casos de (5.9) com os 3 casos de (5.10). No entanto aqui, algumas simplificações podem ser feitas:

- Estarmos no caso  $|k_1 y + k_2 l| < 1$  implica que  $l < \frac{1}{k_2}$  para  $y > 0$  e  $l > -\frac{1}{k_2}$  para  $y < 0$ . Desta forma, sempre que  $\varphi = -\arcsin(k_1 y + k_2 l)$  então, forçosamente  $\dot{l} = y$
- Se  $\varphi = \frac{\pi}{2}$ , então o caso  $l > \frac{1}{k_2}$  e  $y > 0$  para  $\dot{l}$  nunca pode acontecer, já que  $k_1 y + k_2 l < -1$
- De um modo análogo ao anterior, para  $\varphi = -\frac{\pi}{2}$  o caso  $l < \frac{1}{k_2}$  e  $y > 0$  também nunca ocorrerá

De seguida vai-se analisar todos os casos pertinentes para justificar que a função de Lyapunov escolhida é de facto apropriada para provar a estabilidade do sistema em todo o seu domínio.

#### 5.1.1.1 Caso 1

Consideremos as situações em que  $|k_1 y + k_2 l| < 1$  e que corresponde uma variável de controlo  $\varphi = -\arcsin(k_1 y + k_2 l)$ . Neste caso concreto, e tal como já foi explicado anteriormente,  $\dot{l}$  só pode tomar uma forma por imposição de design:  $\dot{l} = y$ . Avaliemos então  $\dot{V}(x)$ .

$$\dot{V}(x) = y\dot{y} + (k_2 l u - c_y)\dot{l}$$

$$\begin{aligned}
&= y [u \sin(-\arcsin(k_1 y + k_2 l)) + c_y] + (k_2 l u - c_y) \dot{y} \\
&= -u k_1 y^2
\end{aligned}$$

Uma vez que tanto  $u$ , como  $k_1$  e  $y^2$  são sempre positivos, então  $\dot{V}(x) \leq 0$ , para qualquer  $y$ .

### 5.1.1.2 Caso 2

O segundo caso de análise acontece para as situações em que  $k_1 y + k_2 l < -1$ , situações essas em que  $\varphi = \frac{\pi}{2}$ . Em concreto, neste caso será analisado o que acontece sempre que  $\dot{l} = 0$  para  $l < -\frac{1}{k_2}$ . Nesta situação vem:

$$\begin{aligned}
\dot{V}(x) &= y \dot{y} + (k_2 l u - c_y) \dot{l} \\
&= y \left[ u \sin\left(\frac{\pi}{2}\right) + c_y \right] \\
&= y(u + c_y)
\end{aligned}$$

Como se sabe que  $u > |c_y|$ , então  $\dot{V}(x) \leq 0$  já que  $y \leq 0$

### 5.1.1.3 Caso 3

O terceiro caso de análise acontece também para as situações em que  $k_1 y + k_2 l < -1$ , situações essas em que  $\varphi = \frac{\pi}{2}$ . No entanto, ao contrário do que acontece no caso 2, neste caso será analisado o que acontece quando  $\dot{l} = y$ , para as situações não incluídas nos casos 1 e 2. Nesta situação, vem:

$$\begin{aligned}
\dot{V}(x) &= y \dot{y} + (k_2 l u - c_y) \dot{l} \\
&= y \left[ u \sin\left(\frac{\pi}{2}\right) + c_y \right] + (k_2 l u - c_y) y \\
&= y u (1 + k_2 l)
\end{aligned}$$

Considerando os casos acima apresentados, há a analisar quatro diferentes situações, consoante o sinal de  $y$  e  $l$

- $0 < l < \frac{1}{k_2}$

Como  $k_1 y + k_2 l < -1$ , o valor máximo que  $y$  irá tomar, de acordo com a variação de  $l$  será  $y = -\frac{1}{k_1}$  para  $l = 0$  e  $y = -2$  para  $l = \frac{1}{k_2}$ . Desta forma está garantido que  $y < 0$ . Uma vez que para estes valores de  $l$ ,  $(1 + k_2 l) > 0$ , então garante-se que  $\dot{V}(x) \leq 0$

- $-\frac{1}{k_2} < l < 0$

Como  $k_1y + k_2l < -1$ , o valor máximo que  $y$  irá tomar, de acordo com a variação de  $l$  será  $y = -\frac{1}{k_1}$  para  $l = 0$  e  $y = 0$  para  $l = -\frac{1}{k_2}$ . Desta forma está garantido que  $y < 0$ . Uma vez que para estes valores de  $l$ ,  $(1 + k_2l) > 0$ , então garante-se que  $\dot{V}(x) \leq 0$

- $l > \frac{1}{k_2}$  e  $y < 0$

Nesta situação específica,  $\dot{V}(x)$  já é garantidamente negativo. Há pois que garantir  $(1 + k_2l)$  é positivo. É de facto fácil de ver que para  $l > \frac{1}{k_2}$ ,  $(1 + k_2l)$  é sempre positivo e, portanto,  $\dot{V}(x) \leq 0$

- $l < \frac{1}{k_2}$  e  $y > 0$

Ao contrário da situação anterior, esta situação ocorre para situações em que  $y$  é positivo. Neste caso é necessário, portanto, garantir que  $(1 + k_2l) < 0$ . Também aqui é fácil de ver que tal acontece já que  $l < -\frac{1}{k_2}$  e, portanto, também aqui  $\dot{V}(x) \leq 0$

#### 5.1.1.4 Restantes

Fica ainda por provar que a função  $\dot{V}(x)$  é semi-definida negativamente para  $\varphi = -\frac{\pi}{2}$ , que acontece para  $k_1y + k_2l > 1$ . No entanto, uma análise semelhante aos casos 2 e 3, que não será aqui apresentada, é de facto suficiente para o provar.

Desta forma, encontrou-se uma função de Lyapunov apropriada para a nossa função de controlo (5.3). Isto garante portanto que (5.3) é assintoticamente estável. Mais, uma vez que (5.9) é válido para todo o domínio, garante-se ainda que (5.3) é globalmente assintoticamente estável.

Na prática, o que isto quer dizer é que, mantendo o AUV uma trajectória rectilínea estável por um espaço de tempo suficientemente longo, o ASV irá de facto convergir para a sua trajectória, independentemente das suas posições iniciais. Obviamente que há ainda que garantir que  $v_{AUV} < v_{ASV}$ .

#### 5.1.2 Controlador

Nesta altura é já possível quantificar algumas das constantes usadas até agora. De facto, tal como descrito acima,  $k_1$  está relacionado com  $d$  (5.5). Sendo 10 metros um valor razoável para essa distância, então  $k_1 = \frac{1}{10}$ .

Linearizando (5.3) em torno do seu ponto de funcionamento, e assumindo variações suficientemente pequenas, é possível reescrever esta equação tal como em (5.13).

$$\begin{bmatrix} \dot{y} \\ \dot{l} \end{bmatrix} = \begin{bmatrix} k_1 & k_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ l \end{bmatrix} \quad (5.13)$$

Calculando os valores próprios do sistema, obtém-se o seguinte;

$$\lambda = -\frac{uk_1}{2} \pm \frac{1}{2}\sqrt{u^2k_1^2 - 4uk_2} \quad (5.14)$$

Uma vez que não é desejável uma resposta oscilatória, então os valores próprios de (5.13) terão que ser reais. Para que isso aconteça, é fácil concluir a partir de (5.14) que  $k_2 \geq \frac{k_1^2}{4}u$ .

Se, tal como se disse inicialmente  $k_1 = 0.1m^{-1}$ , então um bom valor para  $k_2$  será  $k_2 = 0.0025$ .

### 5.1.3 Actuação do Veículo

Tal como descrito anteriormente, o controlo do veículo é feito, a partir dos valores de  $X_{act}$  e  $N_{act}$ . Tal como demonstrado em [16], é possível obter o binário de actuação  $N_{act}$  a partir dos valores de *heading* calculados na secção anterior.

De facto,  $N_{act}$  é obtido à custa de um simples controlo realimentado de velocidade do tipo proporcional-derivativo. Considerando que  $\varphi_{ref}$  é constante já que a orientação de uma trajectória rectilínea não varia, que  $\varphi_{auv} = \arctan \frac{v_y}{v_x}$  e que  $\varphi$  é dado por (5.10), vem:

$$N_{act} = k_p(\varphi_{ref} - \varphi_{asv}) - k_d(\dot{\varphi}_{asv}) \quad (5.15)$$

$$\varphi_{ref} = \varphi_{auv} + \varphi \quad (5.16)$$

Em relação ao valor dos parâmetros  $k_p$  e  $k_d$  do controlador, estes foram inicialmente estimados empiricamente. De facto, como se torna difícil obter um modelo para o heading do veículo, o conhecimento profundo no seio do OSG não só do comportamento do ASV bem como das suas características e de controladores já existentes, permitiu avançar com valores para estas constantes.

$$\begin{aligned} k_p &= 20 \\ k_d &= 20 \end{aligned} \quad (5.17)$$

## 5.2 Controlo de Velocidade

A opção inicial para o controlo de velocidade foi adaptar um dos controladores já existentes, que não é mais do que um controlador de velocidade realimentado do tipo PID [16].

$$X_{act} = k_p(v_{auv} - v_{asv}) + k_i \int (v_{auv} - v_{asv})dt + k_d(s_{auv} - s_{asv}) \quad (5.18)$$

No entanto, cedo se conclui que o desempenho de um controlador deste tipo não se adequava às necessidades. Por conseguinte, posteriormente foi concebido um novo controlador de dois níveis, tal como o esquematizado na figura 5.2.

A ideia fundamental deste controlador é que o controlo da força exercida pelos motores,  $X_{act}$  seja dependente não só da diferença de posições entre AUV e ASV, mas também entre a diferença de velocidades entre eles. O ASV envia periodicamente para o nosso controlador



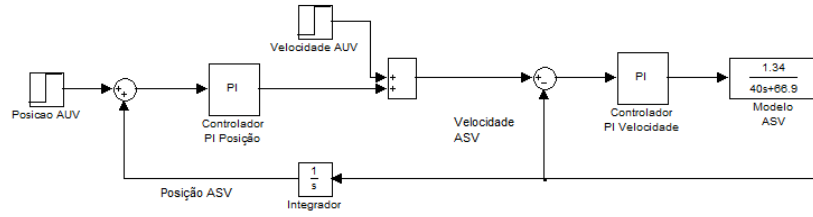


Figura 5.2: Controlador de Velocidade do ASV

os valores da sua velocidade,  $v_{asv} = \sqrt{v_x^2 + v_y^2}$  e da sua posição  $(x, y)$ . Com estes dados, e com os dados fornecidos pelo estimador de mínimos quadrados, é possível calcular a projecção da posição do AUV na sua trajectória estimada  $s_{auv}$ , bem como a projecção da posição do ASV nessa mesma trajectória,  $s_{asv}$ .

$$\begin{aligned} s_{auv} &= \frac{x_0 v_x + y_0 v_y}{v_{auv}} \\ s_{asv} &= \frac{x v_x + y v_y}{v_{asv}} \end{aligned} \quad (5.19)$$

A diferença entre estas duas posições, depois de passar por um controlo do tipo PI, fornecerá o acréscimo de velocidade que vai ser somado ao valor da velocidade do AUV,  $v_{auv}$ . Uma vez que para o ASV conseguir alcançar o AUV, terá que ter maior velocidade, daqui irá resultar o acréscimo de velocidade necessário a  $v_{auv}$ . Esta será a velocidade de referência,  $v_{ref}$  a fornecer a um simples controlador realimentado do tipo PI, cuja saída será a actuação  $X_{act}$ .

Para fazer um estudo sobre os valores mais apropriados para cada uma das constantes dos controladores PI, utilizados na estratégia de controlo de velocidade acima mencionada, partiu-se das equações dos modelos para veículos aquáticos apresentadas no capítulo 2.

De acordo com *Nomoto* [1], um veículo aquático de superfície pode ser aproximado por um sistema de 1ª ordem. Partindo de (2.2) é possível explicitar  $X_{act}$ , a força total exercida pelos motores, em função de uma série de coeficientes hidrodinâmicos (5.20). Nesta expressão  $X_{\dot{u}}$  representa a massa acrescentada na direcção longitudinal e  $X_u u$  e  $X_{|u|u}|u|u$  representam, respectivamente, os coeficientes de arrasto hidrodinâmico de primeira e segunda ordem.

$$X_{act} = (m - X_{\dot{u}})\dot{u} + X_{|u|u}|u|u + X_u u \quad (5.20)$$

Linearizando a equação (5.20) em torno dum ponto de funcionamento equivalente a uma velocidade de 1m/s, obteve-se o seguinte modelo para o nosso ASV.

$$G_p(s) = \frac{1.34}{40s + 66.8} \quad (5.21)$$

Para obter (5.21) partiu-se ainda do princípio que para um comando  $X_{act} = 50$  produz-se uma força tal nos motores do barco, que imprimirão uma velocidade de 1m/s ao barco.

Procurando aproximar a função de transferência em malha fechada do primeiro *loop* de controlo e, posteriormente calculando os pólos de malha fechada do controlador exterior, de posição para que os pólos deste sejam cerca de 10 vezes mais lentos que o controlador mais interno, foi então possível chegar aos seguintes valores:

$$\begin{aligned}k_{p1} &= 30 \\k_{i1} &= 0.1\end{aligned}\tag{5.22}$$

$$\begin{aligned}k_{p2} &= 0.1 \\k_{i2} &= 0.01\end{aligned}\tag{5.23}$$

Aqui,  $k_{p1}$  e  $k_{d1}$  referem-se ao controlador de velocidade, o controlador interno, enquanto que  $k_{p2}$  e  $k_{d2}$  referem-se aos parâmetros do controlador PD externo.

# Capítulo 6

## Resultados Obtidos

Para avaliar o bom funcionamento e desempenho não só dos algoritmos desenvolvidos, mas das estratégias de controlo adoptadas, foi necessário testar exhaustivamente cada uma das partes que constituem a solução desenvolvida. Neste capítulo são apresentados todos os testes efectuados, comprovando o comportamento dos algoritmos.

O processo de testes é composto por duas fases, sendo a primeira uma fase de testes computacional, na medida em que os testes são levados a cabo com o auxílio de simuladores. A segunda fase de testes consiste em testes em ambiente real.

### 6.1 Simulação

Os testes preliminares foram efectuados com duas ferramentas de software, que se revelaram importantíssimas na verificação e validação dos algoritmos desenvolvidos. Os testes feitos permitiram avaliar todas as fases de desenvolvimento dos algoritmos. Assim, foi avaliada a prestação do filtro de Kalman, do algoritmo dos Mínimos Quadrados Recursivos e ainda do controlo do ASV.

Para isso, foram utilizados dois simuladores, nomeadamente o *WaVeSim* e o *ASV Sim*. O *WaVeSim* um ambiente de simulação que emula, com base em modelos físicos o comportamento de veículos aquáticos, mas também de faróis acústicos em pontos pré-definidos, e que são indispensáveis à implementação de vários processos de navegação acústica. Adicionalmente, são também simuladas as comunicações acústicas entre veículos e faróis, bem como vários tipos de sensores usualmente utilizados neste tipo de veículos [20].

O *ASV Sim*, por seu lado, é um simulador de veículos autónomos de superfície, e procura simular o movimento de veículos deste tipo com base nos comandos que recebe.

O procedimento para a execução dos testes é bastante simples, bastando para isso inicializar ambos os simuladores e, de seguida, correr os algoritmos desenvolvidos. O algoritmo de estimação recebe as informações relativas à troca de sinais acústicos, informação esta enviada pelo *WaVeSim*. Na missão de teste utilizada, simula-se o movimento do AUV ao longo de uma trajectória rectangular.

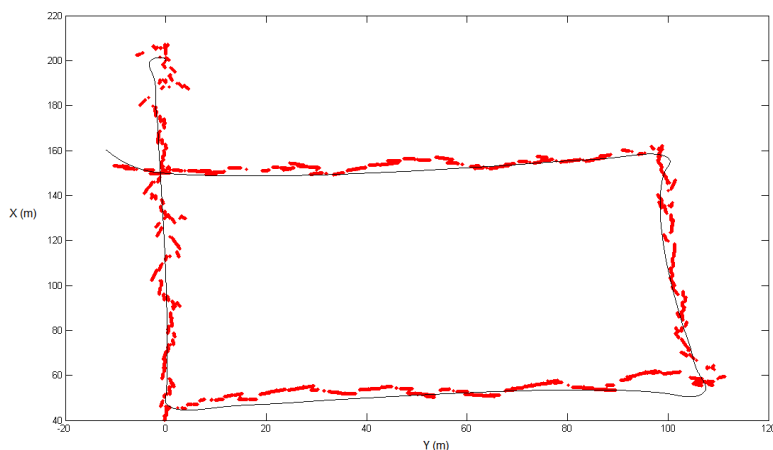


Figura 6.1: Trajectória simulada do AUV (a fino) e trajectória estimada pelo filtro de Kalman (a grosso)

Estimada a posição, este algoritmo envia para o algoritmo de guidance informação relativa à posição do AUV. Com base nestas informações, e nas informações recebidas pelo *ASV Sim* com informação relativa às posições e velocidades do AUV, o algoritmo de *guidance* calcula os controlos do ASV de acordo com a estratégia a seguir, enviando-os de seguida para o *ASV Sim*.

Sendo o objectivo final o controlo do ASV, nos testes efectuados, todos os parâmetros foram ajustados e avaliados tendo em vista esse objectivo. Apesar de nas secções seguintes serem apresentados resultados individuais para cada uma das partes do trabalho, a obtenção de resultados foi um processo iterativo, contínuo e global, envolvendo todos os algoritmos simultaneamente.

### 6.1.1 filtro de Kalman

O filtro de Kalman procura, fundamentalmente, obter estimativas para as posições e, indirectamente, da trajectória efectuada pelo AUV. Nessa forma, na figura 6.1 pode-se ver as trajectórias dadas pelo filtro de Kalman, e por dados de Simulação

Como se pode constatar, a trajectória estimada é bastante aceitável, e dentro dos resultados que se pretendia, com erros na ordem dos 10m a 20m. No entanto, nessa figura é difícil de analisar a evolução temporal da posição do veículo. Nas figuras 6.2 e 6.3 é possível comparar essas evoluções temporais.

Utilizando a informação estimada para as velocidades é possível ver que, tal como esperado, estas não são de facto satisfatórias. De facto, tal, como se pode ver nas figuras 6.4 e 6.5, apesar de as velocidades em ambas as componentes seguirem tendencialmente as velocidades simuladas, estas apresentam-se com grande ruído de fundo, comportamento nada desejável.

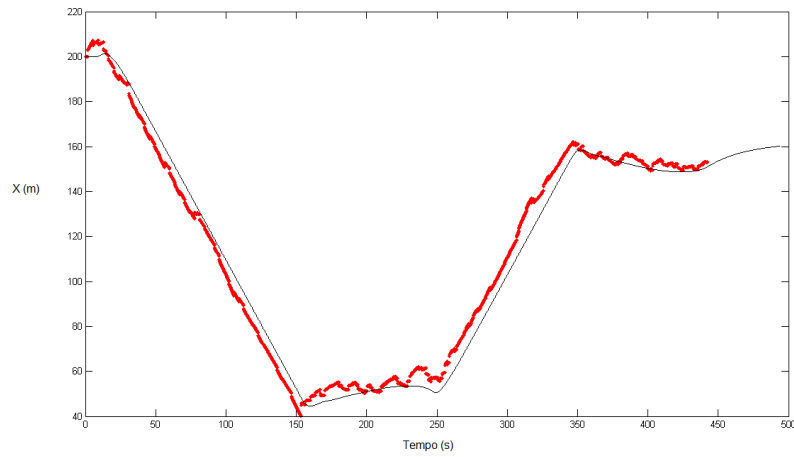


Figura 6.2: Evolução das posições simuladas (a fino) e estimadas pelo filtro de Kalman (a grosso) do AUV ao longo do eixo  $x$

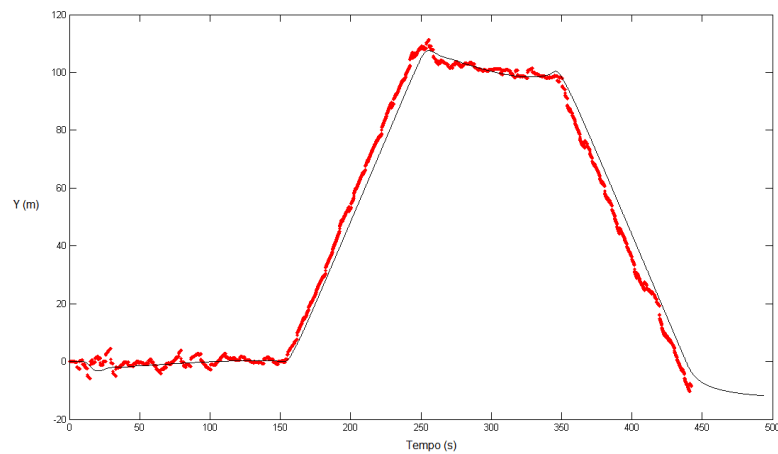


Figura 6.3: Evolução das posições simuladas (a fino) e estimadas pelo filtro de Kalman (a grosso) do AUV ao longo do eixo  $y$

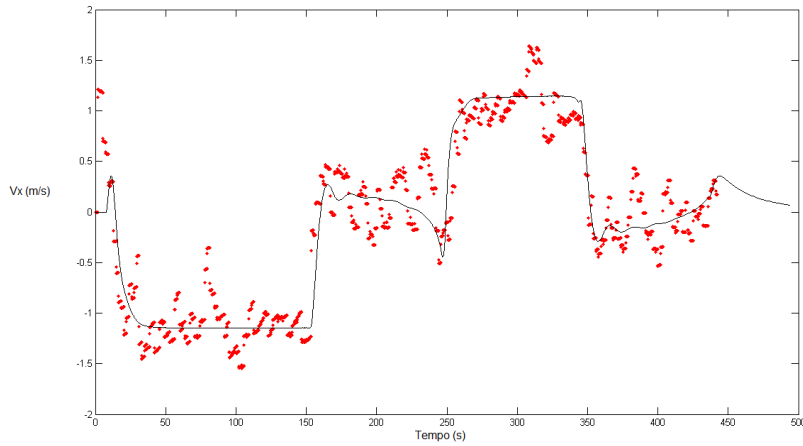


Figura 6.4: Evolução das velocidades simuladas (a fino) e estimadas pelo filtro de Kalman (a grosso) do AUV ao longo do eixo  $x$

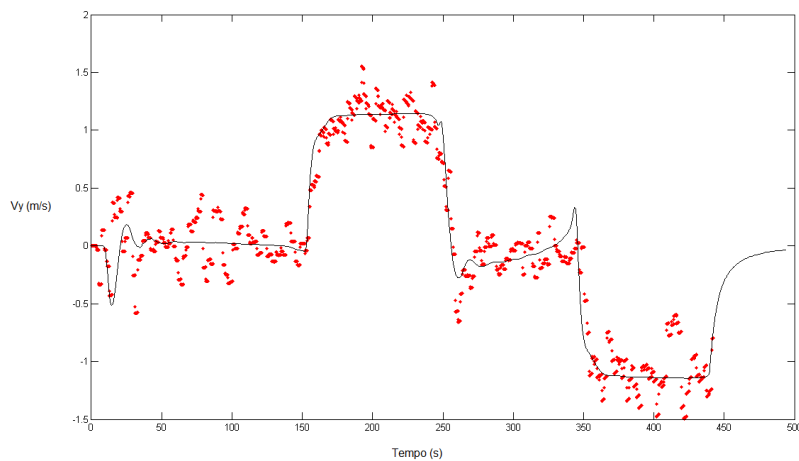


Figura 6.5: Evolução das velocidades simuladas (a fino) e estimadas pelo filtro de Kalman (a grosso) do AUV ao longo do eixo  $y$

Em relação aos parâmetros que podem ser ajustados, nomeadamente dos relativos às covariâncias dos erros associados quer a cada um dos estados, quer à obtenção de novas medidas, o ajuste destes não provoca alterações significativas, já que em ambiente de simulação isso não faz grande sentido. Apesar das virtudes do simulador, este não permite que se simulem condições variadas e adversas para a obtenção de medidas pelo que ajustar estes parâmetros revela-se infrutífero.

Por outro lado, o parâmetro  $\beta$ , incluído no modelo utilizado no filtro, têm alguma influência na obtenção das velocidades, permitindo que estas estimativas sejam mais ou menos contínuas, consoante o seu valor varie entre 1 e 0, respectivamente. No entanto, e uma vez que as velocidades estimadas pelo filtro de Kalman não são relevantes para o estimador de Mínimos Quadrados, não se dedicou grande esforço ao ajuste desse parâmetro. Assim,  $\beta$  foi fixado em 0.02, já que 50 metros é uma distância de paragem considerada razoável para o caso de haver qualquer problema que iniba comunicação com o veículo.

### 6.1.2 Estimador de Mínimos Quadrados

O estimador de Mínimos Quadrados com esquecimento, recebe as posições estimadas pelo filtro de Kalman e obtém, recursivamente, as trajectórias que melhor aproximam um histórico constituído por conjunto de pontos ou posições previamente estimadas. A dimensão desse histórico é determinado pelo valor de  $\lambda$ , tal como detalhado no capítulo 4.

Tabela 6.1: Estimativa de valores de  $\lambda$

| Tempo de atraso | $\lambda$ |
|-----------------|-----------|
| 3.5s            | 0.51      |
| 10s             | 0.80      |
| 20s             | 0.89      |
| 30s             | 0.93      |

Na tabela 6.1 encontram-se os valores para  $\lambda$ , para alguns valores de atraso que se achou convenientes, e calculados tal como indicado em (4.22).

Para avaliar qual o valor de  $\lambda$  mais apropriado a utilizar, ou seja, aquele que minimize a distância entre AUV e ASV ao longo do tempo, fez-se um estudo para estes quatro valores desse parâmetro. Para quantificar, criou-se a seguinte figura de mérito, erro de posição, que procura avaliar isso mesmo.

$$\frac{1}{T} \int_0^T \sqrt{(x_{auv} - x_{asv})^2 + (y_{auv} - y_{asv})^2} dt \quad (6.1)$$

Nas figura 6.6 podemos ver a evolução do erro de posição entre as posições estimadas do AUV e do ASV. Nelas é possível verificar que o erro de posição parece ser minimizado para valores de  $\lambda$  de 0.8 ou 0.89. Aplicando (6.1) a cada uma das séries de dados verifica-se isso mesmo, já que o valor do erro de posição é mínimo para 0.8.

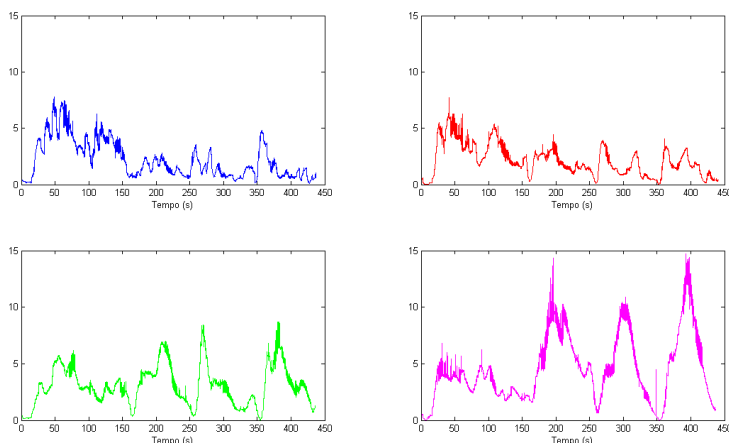


Figura 6.6: Evolução do erro de posição para valores de  $\lambda$  de 0.51 (esquerda em cima), 0.80 (direita em cima), 0.89 (esquerda em baixo) e 0.93 (direita em baixo)

As posições estimadas são de facto as que estão disponíveis, já que na realidade é impossível obter as posições exactas do AUV. No entanto, em simulação, é possível aceder a estas posições bastante para isso aceder aos ficheiros de registo de operações.

Como as posições estimadas e as posições reais não coincidem, seria de esperar que o erro de posição diferisse, aplicando (6.1) aos dados provenientes de ambos os sítios. De facto isso acontece, obtendo-se erros de posição maiores para as posições “reais”.

Os melhores resultados do erro de posição obtidos para posições simuladas foram obtidos para valores de  $\lambda$  entre 0.51 e 0.80, respectivamente 3.5 e 10 segundos. Para estes valores de atraso, a variação do erro de posição, tal como em (6.1) é pouco significativa. De facto, para  $\lambda = 0.51$ , obteve-se um erro de posição com valor de 147.8 enquanto que para  $\lambda = 0.80$  obteve-se um erro de posição com valor de 146.35, o que equivale a uma variação de cerca 1%, valor pouco significativo.

Tomando pois,  $\lambda = 0.8$ , foi possível construir os gráficos das figuras 6.7 e 6.8, que procuram comparar não só as trajectórias, bem como a evolução temporal de posições e velocidades entre os dados “reais”, simulados, para o AUV e os dados estimados pelo algoritmo dos mínimos quadrados recursivos.

### 6.1.3 Algoritmo de Guidance

Em relação ao controlo do ASV Zarco, os objectivos eram otimizar os parâmetros de todos os controladores, procurando maximizar o desempenho total do sistema.

Partindo dos valores apresentados no capítulo 5 foi possível, desde logo, obter resultados animadores. Esses resultados estão visíveis na figura 6.9.

Com alguns ajustes dos parâmetros PD foi então possível melhorar os resultados obtidos. Os melhores resultados que se conseguiu obter estão patentes na figuras 6.10 a 6.12.



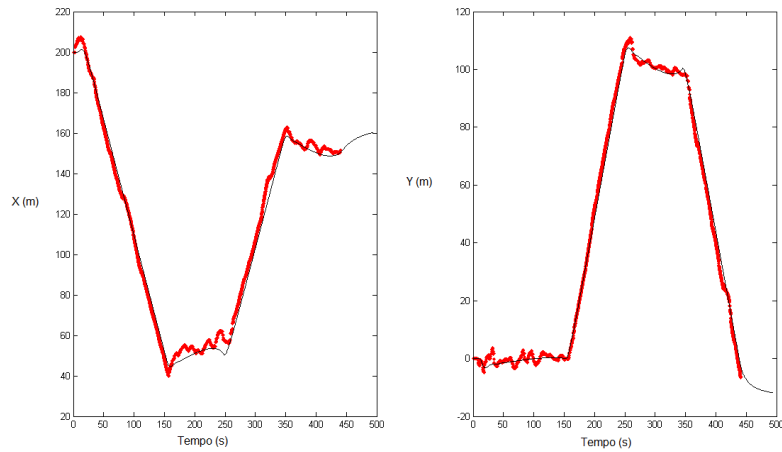


Figura 6.7: Evolução das posições simuladas (a fino) e estimadas pelos Mínimos Quadrados (a grosso) do AUV. À direita  $x$  e à esquerda  $y$

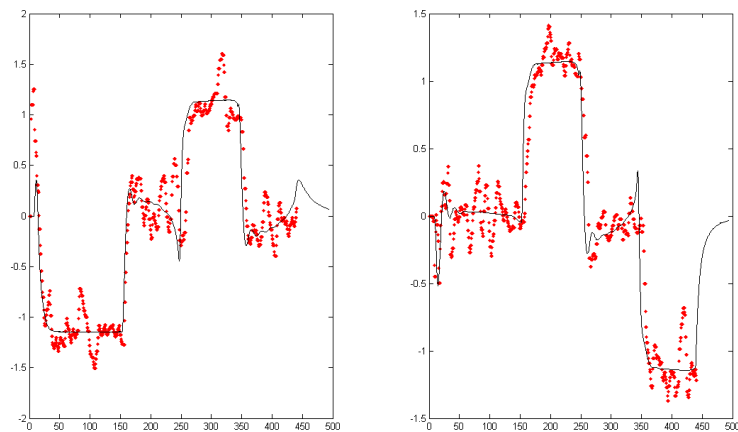


Figura 6.8: Evolução das velocidades simuladas (a fino) e estimadas pelos Mínimos Quadrados (a grosso) do AUV. À direita  $V_x$  e à esquerda  $V_y$

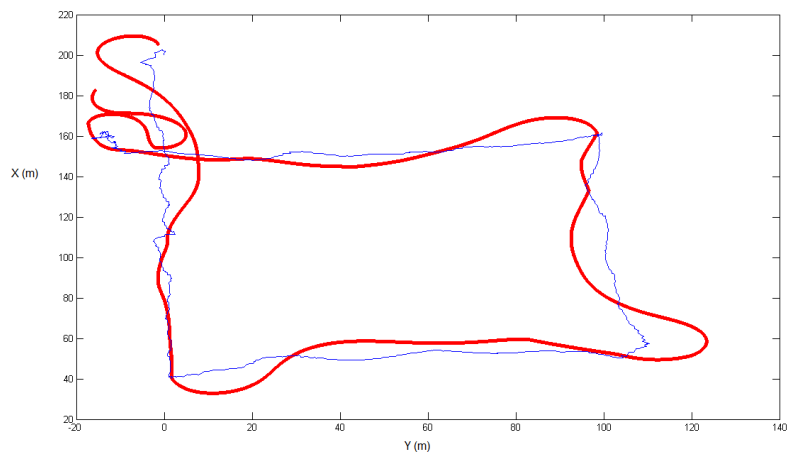


Figura 6.9: Trajéctoria estimada (a fino) e descrita pelo ASV (a grosso)

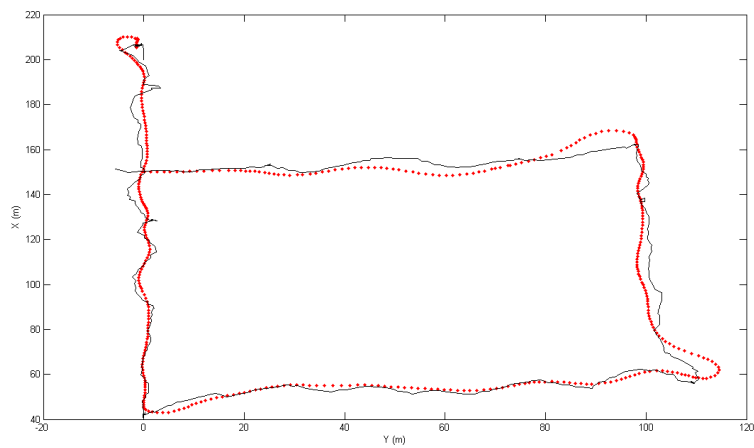


Figura 6.10: Trajéctoria estimada (a fino) e descrita pelo ASV (a grosso)

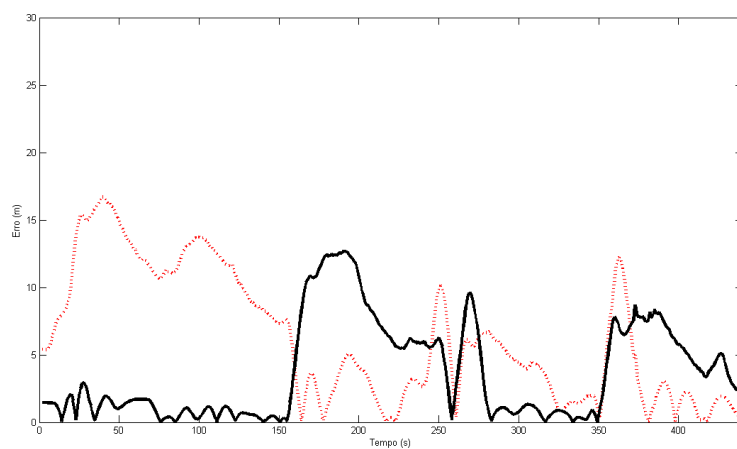


Figura 6.11: Erro de posição em  $x$  (a cheio) e em  $y$  (a tracejado) ao longo to tempo

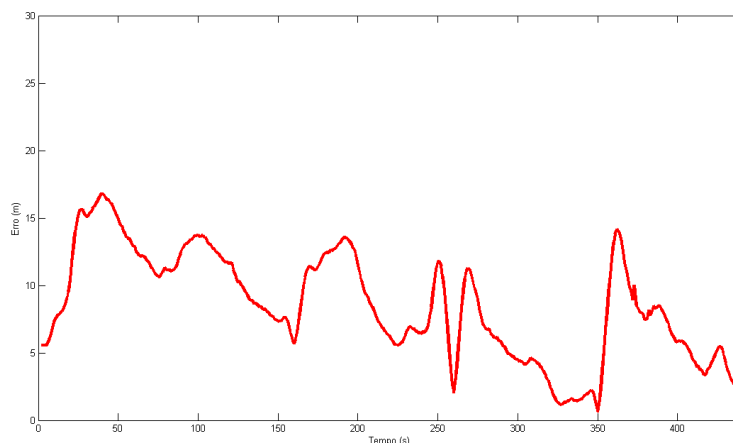


Figura 6.12: Erro total de posição ao longo do tempo

Nas figuras 6.11 e 6.12 vemos a evolução do erro de posição, tanto em termos das suas componentes,  $x$  e  $y$ , como em termos de módulo do erro. Como se poder ver, os erros andam abaixo de 15m, e, na maioria do tempo, abaixo dos 10m. Estes resultados são bastante satisfatórios.

#### 6.1.4 Seguimento de uma recta

Até aqui os testes efectuados basearam-se em hipotéticas missões que poderiam ser executadas, tendo-se verificado que os resultados na sua globalidade são afectados sempre que o AUV muda de direcção. Tendo em conta que todos os algoritmos desenvolvidos foram optimizados segundo a ideia de que, regra geral, o AUV segue uma trajectória rectilínea, será relevante avaliar o algoritmo para missões deste tipo.

As figuras 6.13, 6.14 e 6.15 ilustram o comportamento dos algoritmos desenvolvidos para uma missão desse tipo: no instante  $t = 0$  o ASV encontra-se na posição (205, 0) e o AUV na posição (150, 0), seguindo este uma trajectória rectilínea e uniforme com  $v_x = 0.5m/s$  e  $v_y = 0.3m/s$ .

Tal como se pode ver nas ditas figuras, a solução desenvolvida tem resultados bastante satisfatórios. Após um *overshoot* inicial nas posições, devido ao facto de inicialmente o ASV estar virado a Norte, pode-se ver que ao fim de cerca de 150 segundos o ASV segue a trajectória do AUV com uma precisão assinalável e sem atrasos temporais significativos, o que era de facto um dos resultados pretendidos.

## 6.2 Testes em Ambiente Real

Os testes em ambiente real são testes efectuados com ambos os veículos na água. Habitualmente, as missões de teste do OSG são levadas a cabo no rio Douro, junto à barragem de Crestuma. A escolha por este sítio prende-se com as características do leito do rio nesta

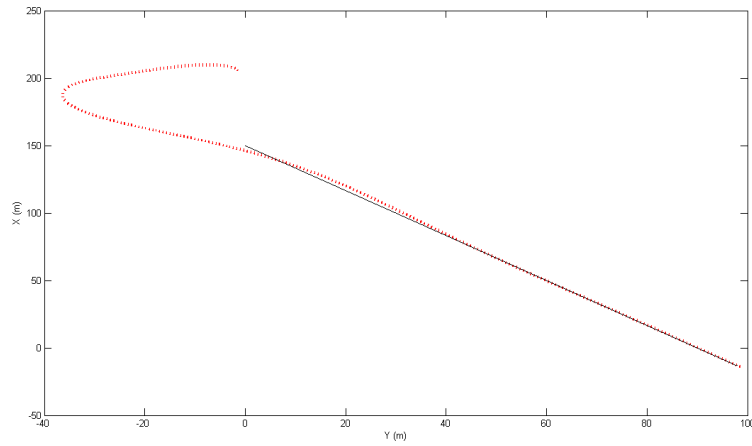


Figura 6.13: Trajectória do AUV (a cheio) e trajectória do ASV (a tracejado) para uma trajectória recitilínea e uniforme

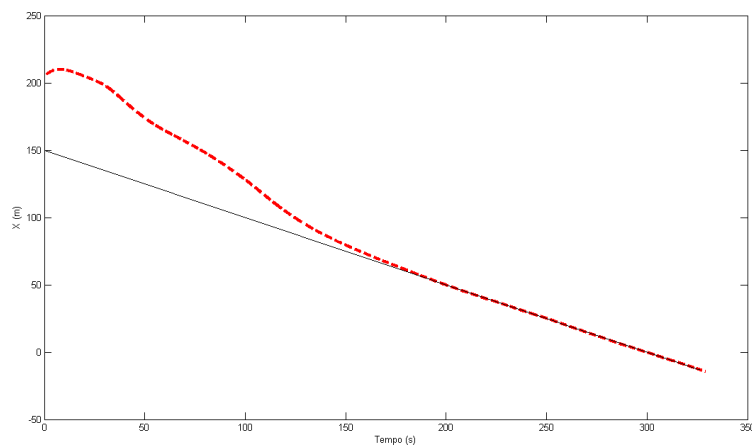


Figura 6.14: Posições do AUV (a cheio) e do ASV (a tracejado) ao longo do tempo, relativas ao eixo  $X$  para uma trajectória recitilínea e uniforme

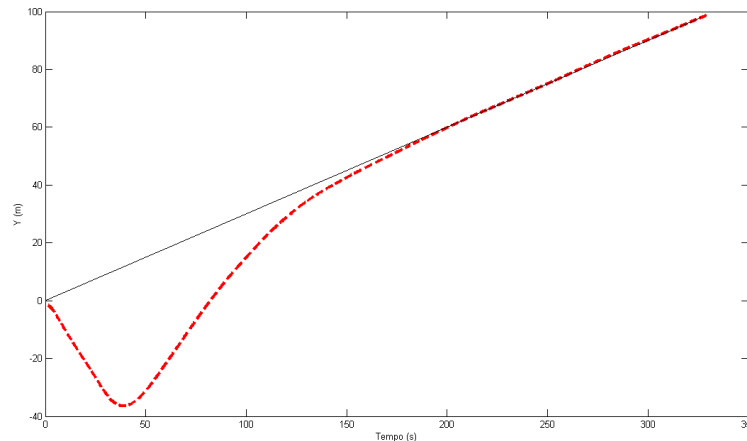


Figura 6.15: Posições do AUV (a cheio) e do ASV (a tracejado) ao longo do tempo, relativas ao eixo  $Y$  para uma trajectória recitilínea e uniforme

zona, já que, regra geral, as águas estão suficientemente calmas para que seja possível fazer todos os testes com segurança.

Os testes do OSG envolvem um grande esforço logístico, já que é necessário transportar para o local dos testes não só os veículos, como também as bóias, e todos os equipamentos de apoio necessários. O mau funcionamento de algum equipamento é também acautelado, existindo, sempre que possível, equipamento adicional de *backup*.

Os testes em ambiente real são também bastante demorados, já que, para além de todo o tempo despendido na preparação e transporte do equipamento, uma vez no local é necessário fazer uma série de procedimentos com vista à calibração de vários equipamentos.

Programar uma sessão de testes é, portanto, algo bastante complexo, pelas dificuldades de várias ordens que todos este processo acarreta. No decorrer deste trabalho, desde logo houve interesse em verificar em ambiente real os resultados alcançados. Na verdade, foi até testada a integração com todo o software já existente, como o software de bordo do ASV Zarco ou o Repositório.

Apesar disso, nem tudo correu como planeado e não foi possível, por razões alheias ao desenvolvimento do trabalho efectuar todos os testes necessários à validação dos algoritmos desenvolvidos. Dessa forma, só é possível, de momento, apresentar resultados experimentais para o algoritmo estimador. Contudo estes testes em ambiente real foram de extrema importância já que permitiram detectar e corrigir uma série de pequenas falhas, impossíveis de detectar por simulação.

O *WaVeSim* não permite a simulação de uma série de fenómenos, como variações no nível de ruído, reflexões das ondas acústicas e as consequentes múltiplas detecções de um mesmo sinal, e que ocorrem sempre que se lida com a emissão de ondas acústicas subaquáticas. Dessa forma, foi impossível prever o comportamento do algoritmo Estimador para a ocorrência de múltiplas detecções de um sinal. De facto, com estes testes experimentais

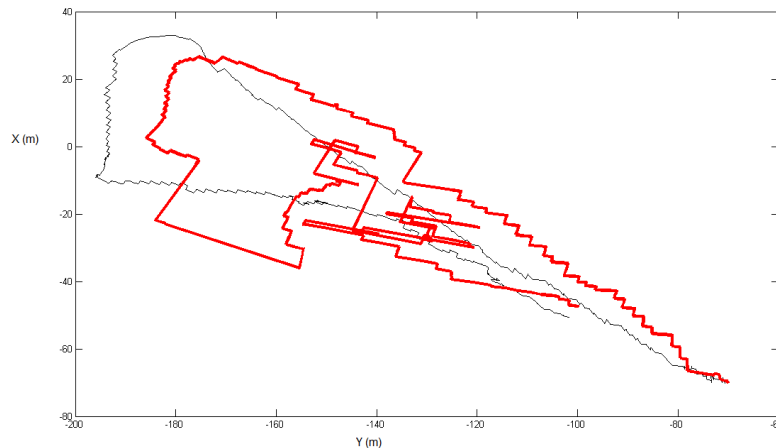


Figura 6.16: Trajectória do AUV: sistema de navegação do AUV (a fino) e algoritmo estimador (a cheio)

foi possível ver claramente que a existência de reflexões dos sinais acústicos emitidos faz com que um mesmo sinal acústico possa ser detectado mais que uma vez. Isto acontece frequentemente e sempre que os sinais acústicos são reflectidos quer pelas margens quer pelo fundo.

Por outro lado, a realização destes testes experimentais permitiu ainda o ajuste de uma série de parâmetros do filtro de Kalman, ajuste esse impossível de realizar só em ambiente simulado. Estes ajustes, fundamentalmente das covariâncias do erro de cada um dos estados do filtro e dos erros associados à obtenção de uma nova medida, dependem das particularidades intrínsecas ao meio e provocam a existência de mais ou menos ruído, de acordo com o local de operação.

### 6.2.1 Algoritmo Estimador

O algoritmo Estimador é responsável por estimar a posição do AUV através da interpretação dos sinais acústicos que são transmitidos quer pelo AUV quer pelas bóias. Os testes que permitem avaliar o seu desempenho consistem, basicamente, em comparar os dados produzidos por este algoritmo, com os dados internos do sistema de navegação do AUV.

Na figura 6.16 é possível comparar, para uma dada missão, a trajectória do AUV com base nos dados do sistema de navegação do AUV por oposição à trajectória do AUV estimada pelo filtro de Kalman. Nesta missão em questão, o AUV parte da posição  $(-70, -70)$ , para onde converge no fim, e as boias estão localizadas em  $(-104,46)$  e  $(116,-13)$ .

É possível ver na figura 6.16 que inicialmente as estimativas são coincidentes, mas que depois evoluem paralelamente ou seja, existe um erro de *offset* entre ambas as trajectórias. Este erro de *offset*, que se cifra entre os 10m ou 15m, tem uma evolução razoavelmente constante até o AUV chegar às vizinhanças da posição  $(0, -180)$ . Este erro de *offset*,

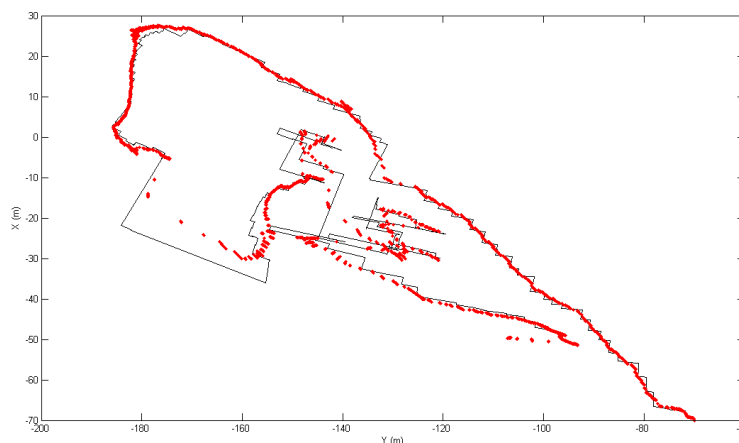


Figura 6.17: Estimativas do AUV dada pelo filtro de Kalman (a fino) e dada pelo estimador de Mínimos Quadrados (a grosso)

apesar de estar dentro dos valores esperados é inerente à própria comunicação acústica e tem justificação quer na geometria da trajectória escolhida, com o AUV a navegar muito próximo de uma das bóias, ficando bastante sensível ao ruído, já que os erros cometidos pelas várias aproximações se tornam bastante mais significativos. Outros factores, como por exemplo a existência de um pequeno *jitter* em  $\delta_X$  e  $\delta_Y$ , podem afectar o resultado já que neste caso, variações de 1ms nestes dois tempos provocam erros de 1 metro na medição de distâncias que são amplificados pela proximidade do AUV às boias.

A partir da posição (0, -180), a estimação do AUV apresenta resultados não tão positivos, justificados em larga medida pela detecção errada da distância entre o AUV e cada uma das bóias, e que se prende com a existência de um grande ruído, que perturba o normal funcionamento do algoritmo.

Comparando os dados do filtro de Kalman com os dados do estimador de Mínimos quadrados, na figura 6.17 é possível verificar que, tal como esperado, o estimador de mínimos quadrados suaviza ligeiramente os dados produzidos pelo filtro de Kalman. No entanto, esta suavização não é suficiente para se obter um conjunto de medidas satisfatórias.

Para obter resultados mais satisfatórios, optou-se então por ajustar os parâmetros do filtro de Kalman, de tal forma que este rejeitasse as medidas erradas que prejudicaram os testes anteriores. Diminuindo o ruído do processo associado às posições,  $q_x$  e  $q_y$ , diminuindo assim a confiança relativa nas medidas de distância, e apertando ainda mais o filtro, diminuindo  $\gamma$ , foi possível obter resultados bem mais satisfatórios, ilustrados nas figuras 6.18 a 6.20

Na figura 6.18 é possível ver a trajectória estimada do AUV após ajuste dos parâmetros. De facto, e apesar de continuar a existir um erro de *offset* ainda na ordem dos 10 a 15 metros, já se consegue obter uma trajectória coerente com a obtida pelo sistema de navegação do AUV. Tal foi conseguido à custa do referido ajuste de parâmetros. Em

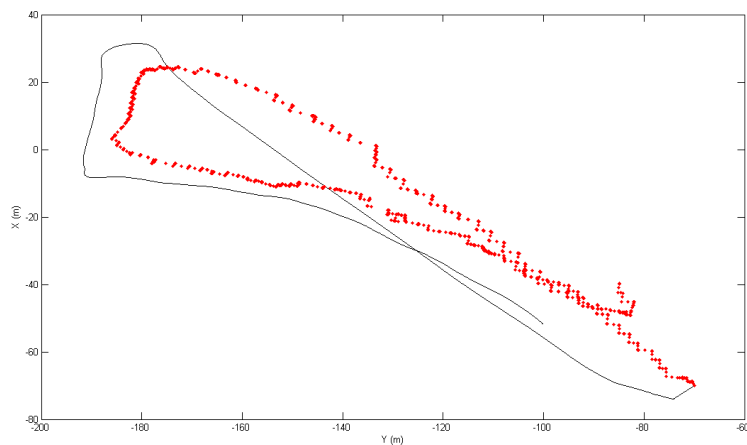


Figura 6.18: Trajectória do AUV: sistema de navegação do AUV (a fino) e algoritmo estimador (a cheio)

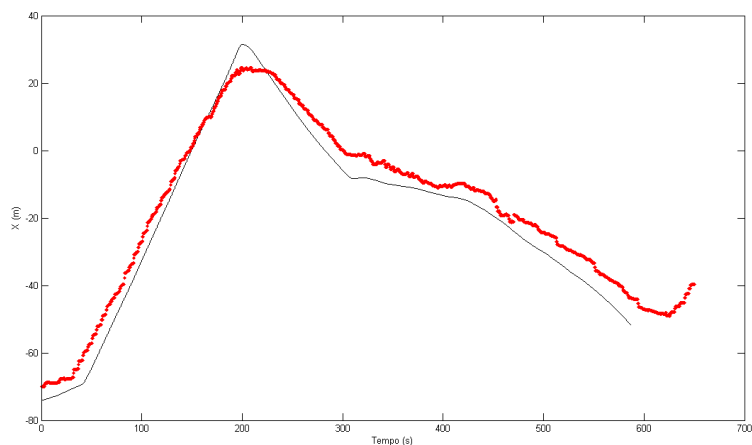


Figura 6.19: Evolução da posição segundo o eixo  $X$  ao longo do tempo



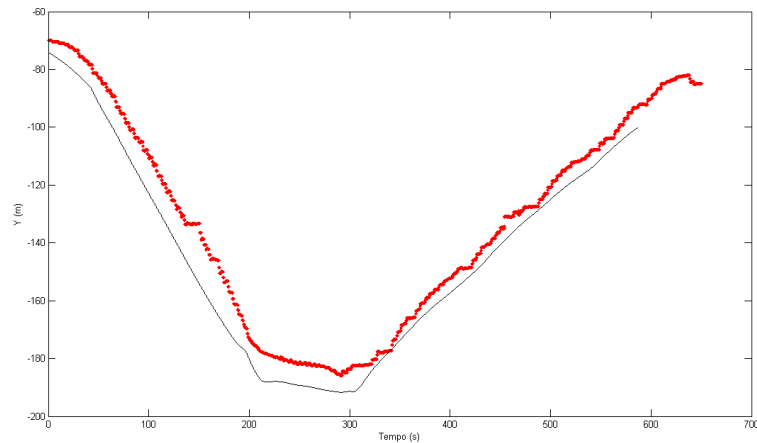


Figura 6.20: Evolução da posição segundo o eixo  $Y$  ao longo do tempo

detalhe, é possível ver nas figuras 6.19 e 6.20 a evolução da posição do AUV ao longo de cada um dos eixos. Nestas figuras é ainda mais claro o erro de *offset* entre a trajetória estimada e dada pelo sistema de navegação, sendo no entanto bastante fácil de verificar que, a menos de este erro, as posições estimadas são bastante satisfatórias.

Partindo destes dados, o Estimador de Mínimos Quadrados consegue ainda obter uma trajetória estimada ainda mais suave, como se pode ver na figura 6.21. Por outro lado, as velocidades ao longo do tempo estimadas por desta forma podem ser vistas na figura 6.22

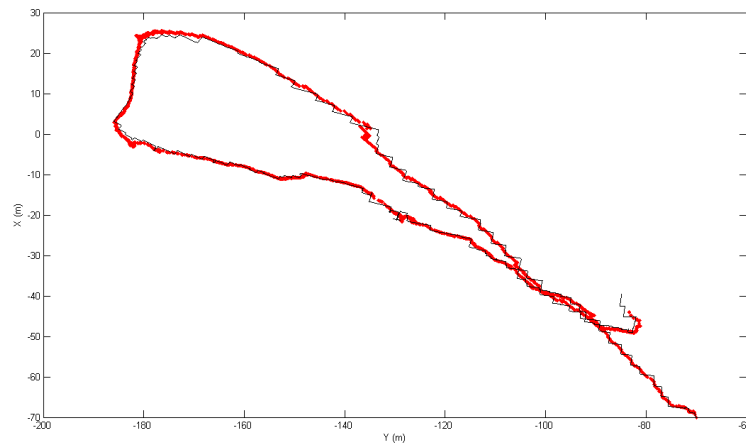


Figura 6.21: Trajectória do AUV: filtro de Kalman(a fino) e estimador de Mínimos Quadrados (a cheio)

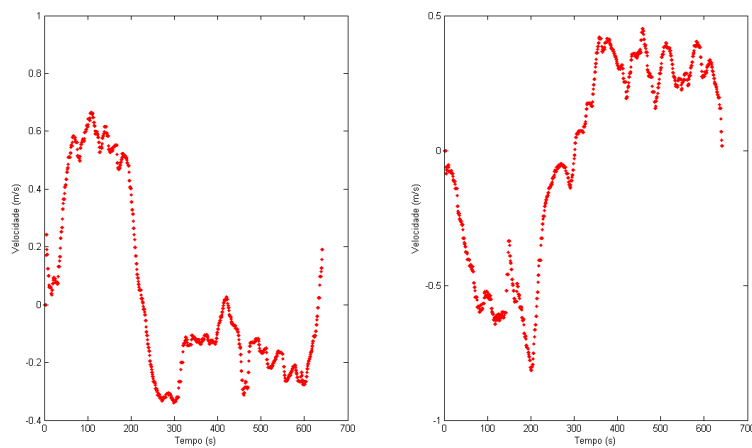


Figura 6.22: Evolução das velocidades ao longo do eixo  $X$  (à esquerda) e  $Y$  (à direita) ao longo do tempo

# Capítulo 7

## Conclusões

### 7.1 Resultados

Os resultados apresentados podem ser considerados bastante satisfatórios, dados os objectivos previstos e as expectativas para o desempenho do sistema. Apesar de tudo, e mesmo não esquecendo a validade dos dados obtidos com recurso aos simuladores, já que estes são bastante completos, este trabalho peca pela falta de alguns resultados em ambiente real que permitam avaliar o trabalho na sua globalidade.

No entanto, e partindo dos resultados que foi possível obter, estes são francamente positivos. Na verdade, as expectativas foram claramente atingidas. Em relação à trajectória percorrida pelo ASV, pode-se dizer que se encontra bastante próxima da executada pelo AUV. No entanto, se se considerar a evolução das posições ao longo de tempo, verifica-se que não existe uma coincidência entre as posições do AUV e do ASV, tanto em relação ao eixo  $x$  como ao eixo  $y$ .

Este facto tem, no entanto, uma explicação, e prende-se com a utilização de um modelo que prevê que os veículos descrevam trajectórias rectilíneas. Isto provoca um ligeiro *overshoot*, visível na figura 6.10, sempre que as trajectórias mudam de posição. Por outro lado, também se pode ver nas figuras 6.11 e 6.12 que enquanto os veículos descrevem trajectórias rectilíneas, a diferença de posições entre AUV e ASV tende a diminuir, sendo até de esperar que, para distâncias superiores, a diferença entre estas distâncias se torne residual.

De salientar ainda que, dada a relevância do trabalho, foi também elaborado um pequeno *extended abstract*, do trabalho desenvolvido e dos resultados esperados, que foi submetido e já aceite para apresentação na conferência *Oceans08 - Quebec*. A *Oceans* é uma conferência realizada duas vezes ao ano e patrocinada pela *Oceanig Engineering Society* do IEEE, e onde são apresentados os novos desenvolvimentos nas áreas de tecnologia, engenharia e ciência do oceanos. Em anexo, pode ser consultada uma cópia do documento submetido.

## 7.2 Trabalho Futuro

Apesar de todo o trabalho desenvolvido, com resultados francamente positivos, este não se esgota aqui. De facto, há uma grande variedade de soluções que a partir daqui podem ser implementadas, podendo mesmo dizer-se que, com este trabalho, uma nova oportunidade se abre.

Numa perspectiva a curto prazo, o que ficou por fazer foi, sem dúvida alguma, a apresentação de resultados globais experimentais obtidos em ambiente real, facto que valorizaria bastante o trabalho. No entanto, num futuro mais longínquo há, com certeza, vários caminhos que podem ser percorridos, e que estarão muito dependentes das áreas de investigação a seguir pelo OSG no futuro.

Uma das opções de estudo será por estratégias de controlo alternativas que permitam, melhorar o desempenho das já existentes, fundamentalmente quando o AUV muda de direcção. Optar por estratégias de controlo óptimo que proporcionem um seguimento do veículo com uma maior economia da bateria será certamente uma opção bastante interessante. Por outro lado, pode ainda ser apetecível derivar uma estratégia de controlo óptimo para atingir a posição do AUV no mais curto espaço de tempo.

Contudo, haverá ainda mais alternativas. Se porventura se pretender um seguimento da trajectória com menos atraso, mais instantânea, será também possível pensar em formas alternativas de estimar a posição do AUV e que passarão, por exemplo, pela detecção de sinais acústicos pelo próprio ASV.

## Anexo A

# Implementação Computacional

Neste anexo são referidos alguns aspectos relativos à implementação dos algoritmos que, não sendo fulcrais para o trabalho, representaram um esforço adicional de aprendizagem, nomeadamente no que diz respeito ao uso de *Sockets* e *Threads*.

Os algoritmos desenvolvidos foram implementados em linguagem C++ sendo possível desta forma tirar partido das particularidades da programação orientada a objectos. Estes algoritmos, por sua vez, correm em sistema operativo Linux, estando totalmente integrados com o *software* de bordo já existente do ASV Zarco.

### A.1 Fluxograma Algoritmo Estimador

O algoritmo Estimador, cuja função é estimar a posição, tem um funcionamento tal como o esquematizado em [A.1](#). Este algoritmo espera periodicamente por mensagens UDP com informações relativas à interrogação de cada uma das bóias, informação essa necessária para a estimação da posição do AUV. No caso de as mensagens recebidas terem sido enviadas quer pela boia 1, quer pela boia 2, então é actualizada a máquina de estados respectiva, que permite estimar a distância entre AUV e cada uma das bóias.

### A.2 Fluxograma Algoritmo *Guidance*

O algoritmo de *Guidance* tem duas tarefas distintas: receber as mensagens do ASV Zarco e do AUV MARES com as informações necessárias para que depois seja possível calcular, de acordo com a estratégia de controlo escolhida, as variáveis de controlo. O funcionamento deste algoritmo está esquematizado em [A.2](#).

### A.3 Sockets

Genericamente falando, os Sockets são um método de comunicação entre um programa cliente e um programa servidor através de uma rede de computadores, e consistem, muito

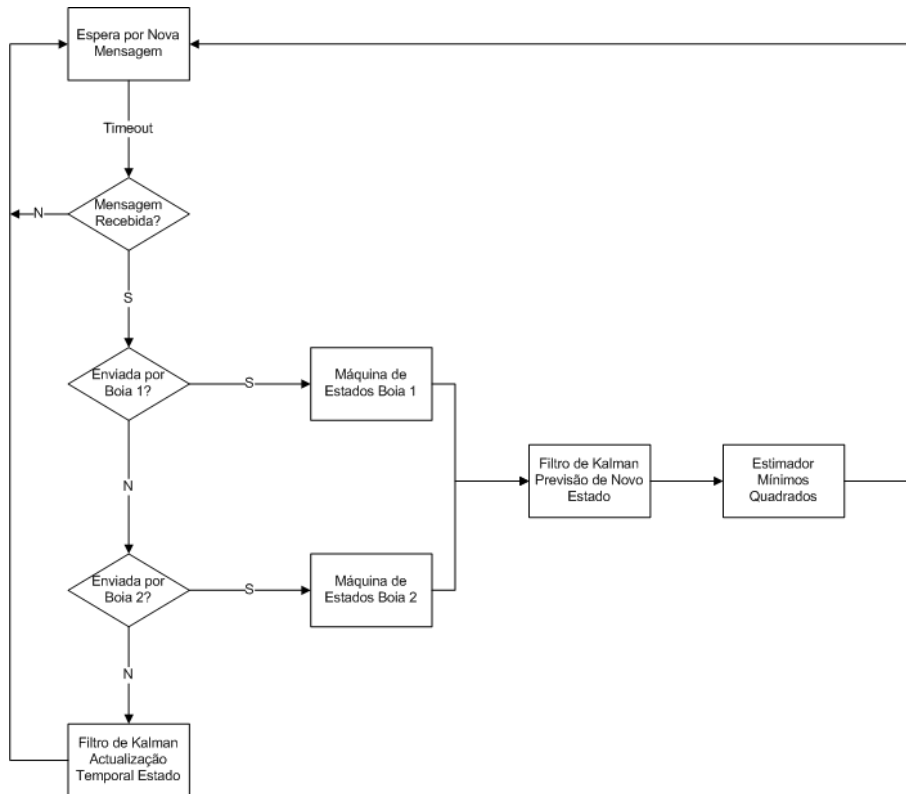


Figura A.1: Fluxograma do algoritmo Estimador

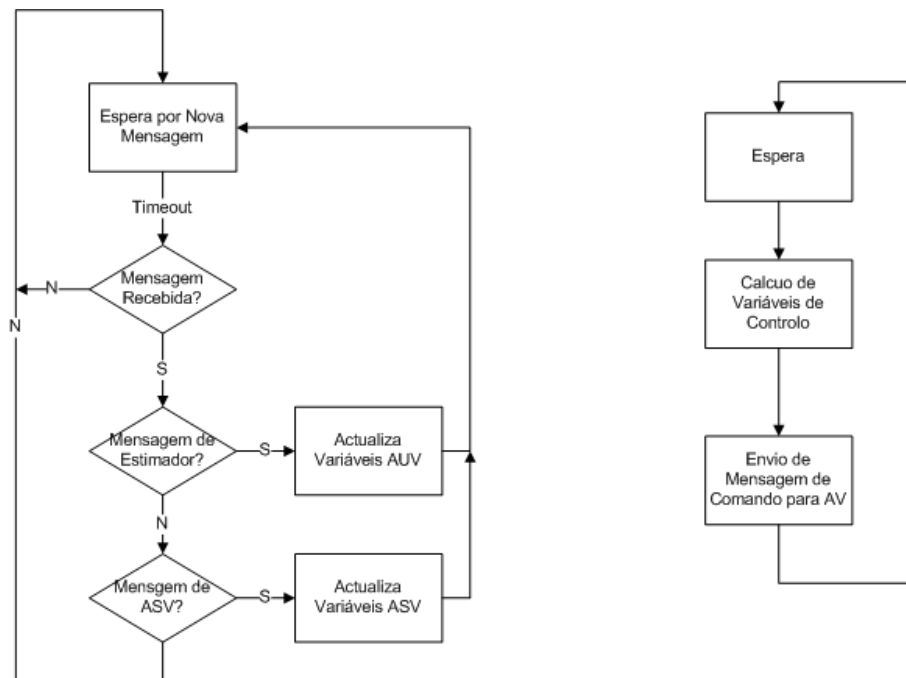


Figura A.2: Fluxograma do algoritmo de *Guidance*

basicamente, numa associação entre única entre um endereço IP e um porto de comunicações.

Este método de comunicação de dados, apesar de ter surgido inicialmente para os sistemas operativos UNIX, encontra-se hoje em dia disseminado, havendo diversas API para os vários sistemas operativos.

Existem 3 tipos diferentes de sockets:

- **Stream Sockets:** Neste tipo de sockets pressupõe-se uma conexão estável entre servidor e cliente, sendo estabelecido um fluxo contínuo de dados entre eles. Apropriado para comunicações orientadas ao canal, ou seja, apropriado para comunicações sobre protocolo TCP.
- **Datagram Sockets:** Neste tipo de sockets, as comunicações são orientadas à mensagem e não se garante, de forma alguma que as mensagens sejam entregues, e que sejam entregues pela ordem em que são enviadas. Este tipo de sockets é apropriado o uso do protocolo UDP
- **Raw Sockets:** Este tipo de sockets é bastante poderoso, já que, apesar de encapsulado no protocolo IP, permite o acesso directo ao seu cabeçalho.

Neste trabalho, foram utilizados *Datagram Sockets* já que as mensagens trocadas entre cada uma das entidades são mensagens UDP, já que não é necessário que se estabeleça qualquer tipo de controlo de fluxo de dados. Desta forma, as mensagens trocadas tanto entre o algoritmo de estimação e o algoritmo de guidance, bem como entre estes e, por exemplo o ASV, é toda implementada através de sockets [21].

Uma sequência típica de inicialização um socket, em linguagem C, é ilustrada em seguida:

```
int sockReceive;
struct sockaddr_in remoteClient;
socklen_t msgAddrLength;

sockReceive = socket(PF_INET,SOCK_DGRAM,IPPROTO_UDP);
remoteClient.sin_family=AF_INET;
remoteClient.sin_addr.s_addr = htonl(INADDR_ANY);
remoteClient.sin_port = htons(ESTIMATOR_PORT);
msgAddrLength=sizeof(remoteClient);

setsockopt(sockReceive, SOL_SOCKET, SO_RCVTIMEO, &timeoutValue, sizeof(timeoutValue));

if (bind(sockReceive,(struct sockaddr *)&remoteClient, sizeof(struct sockaddr))<0) {
cout<<"Erro em bind()"<<'\n';
```

```
}

```

```
receivedBytes = recvfrom(sockReceive, &receivedMessage, MSG_LENGTH, 0,
(struct sockaddr *)&remoteClient, &msgAddrLength);

```

## A.4 Threads

As Threads são uma ferramenta que permite ter, dentro de uma mesma aplicação, várias tarefas distintas a serem executadas paralela e simultaneamente, com os óbvios ganhos de performance e de facilidade de desenvolvimento daí inerentes. Desta forma, uma thread pode-se definir por ser uma sequência de instruções que vão ser executadas num programa, muitas vezes apelidada de função, e de forma independente.

De um modo muito simplista, as threads permitem que os benefícios do multiprocessamento, em que várias tarefas são processadas em simultâneo em processadores distintos, sejam possíveis em máquinas que possuem apenas um processador. No entanto, nestas o multiprocessamento é conseguido através de multiplexagem dos tempos de processamento. Deste modo, o processador alterna rápida e ciclicamente entre cada tarefa em execução, dando a ilusão de simultaneidade de processamento.

Para o algoritmo de guidance, uma vez que há duas tarefas a serem executadas em simultâneo, a saber a recepção de mensagens do estimador e do ASV, e o cálculo e posterior envio para o ASV das variáveis de comando, optou-se pela criação de duas threads, uma para cada uma destas tarefas [22].

Usando a biblioteca *pthread* a criação de threads é bastante simplificada. Uma sequência típica de inicialização de duas threads, 'computeAct' e 'receiveUDP', em linguagem C, é ilustrada em seguida.

```
pthread_t thread_receiveUDP, thread_computeAct;
pthread_create(&thread_computeAct, NULL, computeAct, NULL);
pthread_create(&thread_receiveUDP, NULL, receiveUDP, NULL);

pthread_join(thread_receiveUDP, NULL);
pthread_join(thread_computeAct, NULL);

```



## Anexo B

# Artigo Submetido

Paralelamente ao trabalho desenvolvido, foi também elaborada um *extended abstract* do mesmo em forma de artigo, submetido à conferencia *Oceans'08 Quebec*, a realizar em Setembro deste ano. Este *abstract* foi aceite, e está disponível nas secções seguintes.

### B.1 Abstract

The following addresses the control of an Autonomous Surface Vehicle (ASV) to follow the trajectory made by an Autonomous Underwater Vehicle (AUV) when the last is performing any given pre-programmed mission. In fact, it has been proved to be of great interest to have an ASV that could follow on the surface and even catch up the trajectory performed by the AUV, when executing a given mission.

In order to achieve this desired coordinated motion between AUV and ASV, it would make sense just to program each of the vehicles with the same mission. However, due to the nature of vehicles, missions and also due to the localization system used, with this kind of solution some problems would arise, namely related with timings and synchronization, which are indeed difficult to overcome.

The solution here proposed tries to estimate the AUV position, by tapping the signals exchanged between the former and each of the beacons of the acoustic localization network, and control and actuate the ASV in accordance.

With the final article, results will be presented, not only supporting our theoretical and mathematical analysis, but also experimental data obtained by some validation tests done in the water. Also, different control strategies will be suggested, according to the objectives of each mission.

### B.2 Introduction

The AUV built and used by the OSGroup, FEUP, uses a two beacon LBL based acoustic network in order to get some positioning information. Prior to any mission, the vehicles are

informed of the actual global coordinates of the two beacons. Then, in order to know its exact localization at any given time, each vehicle has to interrogate each beacon, sending an acoustic signal with a specific frequency and waiting for the beacon's reply. By timing this acoustic events, it is then possible to compute the actual distance of a given vehicle to each of the two beacons and, therefore, its real-time global coordinates.

### B.3 Estimation

The algorithm here presented is based on a algorithm proposed in [15] and assumes that the AUV positions remains stationary between the interrogation of the beacon and the reception of the correspondent answer. It is also considered that the depths the AUV reaches while in mission are quite small relative to the distances to both beacons and, thereby, we can assume only motion in the horizontal plane. Having this in mind, and considering that the AUV moves with approximately constant speed, the state-space model of the AUV motion can be described as below.

$$\begin{bmatrix} \dot{x} \\ \dot{v}_x \\ \dot{y} \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \beta \end{bmatrix} \begin{bmatrix} x \\ v_x \\ y \\ v_y \end{bmatrix} \quad (\text{B.1})$$

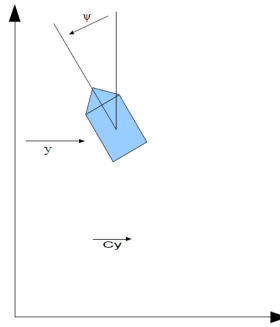
By having an external acoustic receiver listening the acoustic signals in the range of those exchanged between the AUV and the beacons, and with prior knowledge of local sound propagation speed and the delays associated with this communications, it is straightforward to build an algorithm that calculates the distance between the AUV and both beacons.

However our objective is to estimate the global coordinates of the AUV, so we need to estimate the real-time state of the system. This is done by using a continuous-discrete Extended Kalman Filter.

### B.4 Control and Actuation

Our goal is that the ASV can follow the trajectory of the AUV. With the data output by the Kalman Filter, it is relatively easy to build a sequence of straight lines that represents the trajectory the AUV is taking.

However this is, indeed, not enough, as a strategy that enables the ASV to reach the AUV trajectory had to be chosen. For that, we need to know, not only the relative distance from the AUV to the ASV, but also the heading of the ASV relative to the trajectory the AUV is undergoing, always having in mind that the AUV trajectory is build up from a sequence of straight lines.



The main idea is that if the ASV is far enough from the AUV, then it should go straight ahead on a direction perpendicular to the ASV direction on full speed; as it approaches the AUV, it should start changing its heading to one that suits its needs. Based on the track-keeping systems referred in [1], it was possible to derive a simplified kinematics equation to the ASV that synthesizes the desired behaviour. We can see it in (B.2), where  $u$  is the forward speed (surge) and  $C_y$  represents any water current that may exist. It is assumed that  $u > |C_y|$ , meaning this that the actuation is powerful enough to overcome the disturbances.  $\psi$  is our control law and it comes directly from the heading loop control. As we are in presence of disturbances, it is also necessary to take in account an additional state that ensures a null steady state tracking error.

$$\dot{y} = u \sin(\psi) + C_y \quad (\text{B.2})$$

In order to assure that such control would indeed result on the follow up of the desired trajectory, we started by proving the stability of this equation. As the control law is non-linear, care had to be taken, as the usual linear control tools don't apply. According to Lyapunov's direct method, prove that the function is asymptotically stable can be done by finding a proper Lyapunov function [12].

By finding a proper control function, which represents the behaviour desired, and by finding an appropriate Lyapunov function, we were able to determine the global asymptotic stability of the function and, therefore, to study different control strategies.



# Referências

- [1] Thor I. Fossen. *Guidance and Control of Ocean Vehicles*. Wiley, August 1994.
- [2] Thor I. Fossen Tristan Perez. Kinematic models for seakeeping and manoeuvring of marine vessels. *Modeling, Identification and Control*, pages 1–12, 2007.
- [3] T. Perez and T.I. Fossen. Tutorial on modelling and simulation of marine system dynamics. ifac conference on control applications in marine systems (cams). bol, croatia. 2007.
- [4] Xiaoping [1] Yun, Eric R.[2] Bachmann, [1] Suatarslan, Kadir [1] Akyol, and Robert B.[2] McGhee. An inertial navigation system for small autonomous underwater vehicles. *Advanced Robotics*, 15:521–532, September 2001.
- [5] Ivo Alcantara Dias. *High Precision Underwater Positioning*. Relatório de Projecto submetido para satisfacao parcial dos requisitos do grau de licenciado em Engenharia Electrotecnica e de Computadores, Porto, 2007.
- [6] Luis Antonio Dias Madureira. *Sistema de Navegacao Acustica para Multiplos Veiculos Subaquaticos*. Dissertacao submetida para satisfacao parcial dos requisitos do grau de mestre em Engenharia Electrotecnica e de Computadores, Porto, 2004.
- [7] Edwin Olson, John J. Leonard, and Seth Teller. Robust range-only beacon localization. *Oceanic Engineering, IEEE Journal of*, 31:949–958, 2006.
- [8] A. Matos and N. Cruz. Auv navigation and guidance in a moving acoustic network. In *Oceans 2005 - Europe*, volume 1, pages 680–685 Vol. 1, 2005.
- [9] J. Curcio, J. Leonard, J. Vaganay, A. Patrikalakis, A. Bahr, D. Battle, H. Schmidt, and M. Grund. Experiments in moving baseline navigation using autonomous surface craft. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 730–735 Vol. 1, 2005.
- [10] Wen-Hui Cheng. A study of increasing the precision of navigation position for submerged body. *Ocean Engineering*, 31:693–707, April 2004.
- [11] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice*. Prentice Hall, har/dsk edition, February 1993.
- [12] Jean-Jacques Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, 1991.
- [13] Eronini I. Umez-Eronini. *System Dynamics and Control*. CENGAGE-Engineering, 1 edition, July 1998.
- [14] Grupo de Sistemas Oceanograficos Antonio Santos. Modelo de comunicacoes oceansys. 2007.

- [15] Nuno Cruz Anibal Matos. Algorithms for external tracking of an auv. *Proceedings of the Conference IAV04, Lisbon, July 2004*, 2004.
- [16] A. Matos and N. Cruz. Coordinated operation of autonomous underwater and surface vehicles. In *Oceans 2007*, pages 1–6, 2007.
- [17] Lennart Ljung. *System Identification : Theory for the User*. Prentice Hall, har/dsk edition, 1999.
- [18] A. Vahidi, A. Stefanopoulou, and H. Peng. Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. *Vehicle System Dynamics*, 43:31–55, 2005.
- [19] Katsuhiko Ogata. *Modern Control Engineering*. Prentice Hall, 4 edition, November 2001.
- [20] Antonio Manuel Nogueira Santos. *WaVeSim - Ambiente de simulacao para veiculos aquaticos*. Tese submetida a Faculdade de Ciencias da Universidade do Porto para obtencao do grau de Mestre em Informatica, Porto, 2006.
- [21] W. Richard Stevens, Bill Fenner, and Andrew M. Rudoff. *Unix Network Programming, Volume 1: The Sockets Networking API (3rd Edition)*. Addison-Wesley Professional, 3 edition, November 2003.
- [22] Bradford Nichols, Dick Buttlar, Jacqueline Proulx Farrell, and Jackie Farrell. *Pthreads Programming: A POSIX Standard for Better Multiprocessing*. 1 edition, September 1996.