

Sistema de localização relativa de um sistema robótico subaquático

JOSÉ PEDRO ALMEIDA MARQUES DE OLIVEIRA
dezembro de 2021

POLITÉCNICO DO PORTO
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Sistema de localização relativa de um sistema robótico subaquático

José Pedro Almeida Marques Oliveira

Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização em Sistemas Autónomos



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto

Dezembro, 2021

*Esta dissertação satisfaz, parcialmente, os requisitos que constam da Ficha de
Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em
Engenharia Electrotécnica e de Computadores, Área de Especialização em
Sistemas Autónomos.*

Candidato: José Pedro Almeida Marques Oliveira, N.º 1160955,
1160955@isep.ipp.pt

Orientação Científica: Eduardo Alexandre Pereira da Silva, eps@isep.ipp.pt

Coorientação Científica: José Miguel Soares de Almeida, jsa@isep.ipp.pt



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Dezembro, 2021

Agradecimentos

Ao meu coorientador, Professor Eng. José Almeida, por todo o apoio prestado no decorrer do desenvolvimento da tese. Apesar do pouco tempo disponível, sempre se esforçou para ajudar e tirar as dúvidas que iam surgindo. Agradecer também ao Professor Eng. Eduardo Silva pela ajuda disponibilizada.

A todos os amigos que fiz durante o mestrado, um obrigado pelas partilhas e experiências. Em particular a três deles, ao André e ao Zé por estarem sempre disponíveis para ajudar e por todos os projetos que fomos partilhando. Ao David, por ter sido meu parceiro durante estes últimos dois anos, sem ele não estaria a terminar esta etapa com sucesso .

Por fim, agradecer à minha família, por me proporcionarem a possibilidade de seguir uma vida académica e pelo apoio dado não só financeiramente mas também em todos os outros aspetos. À Joana pelo apoio em todas as decisões que tomei nestes últimos meses e pelos seus sinceros conselhos, um muito obrigado.

Resumo

Esta dissertação nasce na necessidade de um sistema de localização para um sistema de docagem. Este sistema de docagem têm a necessidade de se localizar a si num referencial de navegação mas também localização um *lander* robótico, no qual irá docar.

O sistema de localização no referencial de navegação tem como base a fusão de vários sensores, presentes na doca. Na fusão sensorial optou-se por um filtro *Extended Kalman Filter* (EKF) com dados sensoriais de um sistema *Inertial Navigation System* (INS), do *Doppler Velocity Log* (DVL), de um sistema *Ultra Short baseline* (USBL) e de um sensor de pressão. Para além disso, foi estimado um modelo dinâmico idêntico à estação de docagem.

Para a o sistema de docagem conseguir localizar o *lander* robótico foi utilizado uma solução idêntica à anterior. Um filtro onde é utilizado ou as medidas de um sistema USBL ou as medidas de um algoritmo de visão. Neste é utilizado um modelo de velocidade constante.

Para a validação do algoritmo desenvolvido foi utilizado um simulador desenvolvido previamente. Este é capaz de simular a manobra de docagem, permitindo obter *logs* de dados obtidos pelos sensores simulados.

Os resultados obtidos com o sistema de localização são bastante satisfatórios; apresentado erros inferiores a 10 cm na posição e 1 ° na orientação durante a sua missão de docar ao *lander*. Para além dos dados da sua missão, foram realizados testes, como simular falhas de sensores, que provam a robustez do filtro.

Relativamente aos resultados do filtro de localização relativa, tal como referido foram utilizados medidas de duas fontes diferentes. Aqui foi pensado utilizar as medidas obtidas pelo sistema USBL enquanto a câmara não consegue ver o *lander*, a partir do momento que consegue passa a utilizar a câmara, através de um algoritmo de visão, para obter a posição relativa. Quantos aos resultados foram obtidos erros inferiores a 1 cm em posição x e y e inferior a 10 cm em profundidade.

Palavras-Chave: Sistema/Estação de Docagem, Sistema de Localização Subaquático, Fusão Sensorial, Modelo Dinâmico, Filtro EKF, Sistema USBL, DVL, Sistema INS

Abstract

This thesis addresses the need of a localization system for a underwater docking system. Since the docking system is full autonomous, it is necessary to localise itself in his navigation referential and capable of localise an robotic lander, which the docking system will dock.

This thinking led to the development of two localization systems that complement one another. The first one is capable of estimate the state that characterize the docking station, in his navigation referential. While the other one is capable of localizing the lander in the docking referential.

So, the first one fuses the data from several sensors such as: a DVL, a INS system, a pressure sensor and a USBL system. Furthermore, in the filter predict phase a docking system dynamic model is employed.

To localise the lander in the docking referential a filter when a constant velocity model is used to predict lander's position. An vision algorithm, based on a single camera, and a USBL system are used in the update phase.

A simulator is used to validate the two systems. This simulator is capable of reproduce the docking maneuver, allow to obtain logs with data from the sensors simulated.

For the docking localization, the results are satisfying. It is capable of estimate the position with an error of less than 10 cm, and in the orientation estimation, the error is inferior to 1° . In addition, the filter robustness was verified when the data from DVL and USBL had flaws.

Relatively to the lander localization results, two different sources of data are used. When the camera can not see the robotic lander, the data from the USBL is preferable. The output from the vision algorithm are only used when the sensor can see the lights from the lander. That error obtain is less than 1 cm for x and y position and 10 cm for depth.

Keywords:Docking Station, Underwater Localization, Data Fusion, dynamic model, EKF, USBL, DVL, INS

Índice

Lista de Figuras	vii
Lista de Tabelas	ix
Listagens	ix
Lista de Acrónimos	xi
1 Introdução	1
1.1 Contextualização	2
1.2 Motivação e Formulação do problema	4
1.3 Objetivos	5
1.4 Organização da Dissertação	5
2 Estado da Arte	7
2.1 Problemática da localização	7
2.2 Métodos de localização subaquático	9
2.2.1 <i>Dead-Reckoning</i> e Inercial	10
2.2.2 Geofísica	11
2.2.3 Acústica	12
2.2.4 Estimação e Filtragem	14
2.3 Navegação Cooperativa	15
3 Conceitos Teóricos	17
3.1 Referenciais	17
3.1.1 Referencial Inercial (<i>i-frame</i>)	17
3.1.2 Referencial Local (<i>n-frame</i>)	18
3.1.3 Referencial do Corpo (<i>b-frame</i>)	19
3.2 Rotações tridimensionais	20
3.2.1 Ângulos de Euler	20
3.2.2 Quaterniões	21
3.3 Transformação de Referenciais	22
3.3.1 Transformação <i>b-frame</i> para <i>n-frame</i>	22
3.3.2 Transformação entre Referencial Local ENU e NED	22

3.4	EKF	23
3.5	Modelo hidrodinâmico	24
4	Arquitetura do sistema	29
4.1	Sistemas Robóticos integrantes	29
4.1.1	Bóia	29
4.1.2	Lander Robótico Turtle	30
4.2	Localização da doca no referencial de navegação	30
4.2.1	Software	31
4.2.2	Hardware	32
4.3	Localização Turtle relativa à doca	34
4.3.1	Hardware	35
4.3.2	Software	35
5	Implementação	37
5.1	Filtro de Kalman estendido - localização doca	37
5.1.1	Fase de previsão	38
5.1.2	Fase de atualização	40
5.2	Filtro de Kalman estendido - localização doca $\leftrightarrow$ Turtle	43
5.2.1	Fase de previsão	43
5.2.2	Fase de atualização	45
5.3	Modelo dinâmico	46
5.4	Algoritmo de visão	48
6	Resultados	51
6.1	Validação do trabalho desenvolvido	51
6.1.1	Simulação dos sensores	52
6.2	Validação do filtro de Kalman estendido - localização doca	52
6.2.1	Testes suplementar - Ruído em USBL e DVL	58
6.2.2	Testes suplementar - DVL a funcionar a uma certa profundidade	59
6.2.3	Conclusão da validação do filtro	60
6.3	Validação do filtro de Kalman estendido - localização doca $\leftrightarrow$ Turtle	60
6.4	Validação do sistema de visão	63
6.5	Comparação entre filtro do modelo dinâmico com filtro INS	64
7	Conclusões	67
7.1	Trabalho Futuro	68
	Referências	69

Lista de Figuras

1.1	Fase de procura	3
1.2	Fase de aproximação	4
1.3	Fase de docagem	4
2.1	Várias técnicas para localização subaquática, baseado em [1]	10
2.2	Descrição gráfica de cada um dos sistemas: LBL, SBL e USBL [2]	13
3.1	Referencial inercial [3]	18
3.2	Referencial Local NED em relação ao referencial ECEF [4]	19
3.3	Exemplo de um referencial do Corpo (<i>body frame</i>) [5]	19
3.4	Representação dos ângulos de Euler [6]	20
4.1	Arquitetura geral da solução para localização absoluta do sistema aquático	30
4.2	Diagrama do bloco filtro de Kalman estendido da arquitetura do <i>software</i>	31
4.3	<i>Pinger</i> USBL S2C T 18/34 [7]	32
4.4	DVL A50 da WaterLinked [8]	33
4.5	<i>Cube Orange</i> sistema para obtenção dos dados inerciais [9]	33
4.6	Arquitetura de geral da solução para localização absoluta do sistema aquático	34
4.7	Diagrama do bloco filtro de Kalman estendido da figura 4.1	35
5.1	Passos realizados na algoritmo de visão: Imagem retirada da câmara, aplicação preto e branco, e aplicação do <i>threshold</i>	49
6.1	Comparação da trajetória estimada e a verdadeira	53
6.2	Comparação entre a posição estimada e a verdadeiras, em cada um dos três eixos	54
6.3	Inovação da posição em x , y e z	55
6.4	Comparação entre as velocidades estimadas e as verdadeiras, em cada um dos três eixos	55
6.5	Inovação da velocidade em x , y e z	56
6.6	Comparação entre a orientação estimada e a verdadeira, em cada um dos três eixos	57

6.7	Inovação da orientação em ϕ , θ e ψ	57
6.8	Comparação entre a velocidade angular estimada e a verdadeira, em cada um dos três eixos	58
6.9	Comparação entre a posição estimada e a verdadeira, em cada um dos três eixos	61
6.10	Comparação entre a posição estimada e a verdadeira, em cada um dos três eixos, em foco o momento após a primeira informação da câmara	62
6.11	Inovação da posição em x , y e z	62
6.12	Passos realizados no algoritmo de visão: Imagem retirada da câmara, aplicação preto e branco e aplicação do <i>threshold</i>	63
6.13	Vários exemplos do output do sistema de visão	64

Lista de Tabelas

3.1	Variáveis associadas ao estado de um veículo aquático	24
3.2	Lista dos termos e coeficientes do modelo dinâmico	26
5.1	Parâmetros físicos do sistema de docagem	46
5.2	Coeficientes do sistema de docagem	47
5.3	Localização dos LED's no referencial da câmara	48
6.1	Caracterização de diversos parâmetros do acelerómetro e do giroscópio	52
6.2	Quantificação do erro e desvio padrão em posição do sistema de localiza- ção do sistema de docagem	54
6.3	Quantificação do erro e desvio padrão em velocidade do sistema de localização do sistema de docagem	56
6.4	Quantificação do erro e desvio padrão em orientação do sistema de localização do sistema de docagem	56
6.5	Quantificação do erro e desvio padrão em velocidade angular do sis- tema de localização do sistema de docagem	58
6.6	Quantificação do erro e desvio padrão dos estados para o teste referido	59
6.7	Quantificação do erro e desvio padrão dos estados para o teste referido	60
6.8	Quantificação do erro e desvio padrão da posição do sistema de localiza- ção entre o Turtle e a doca, após a primeira informação da câmara	61
6.9	Erro máximo entre a posição verdade e obtida pelo algoritmo de visão	64
6.10	Quantificação do erro médio para os doze estados nos dois filtros di- ferentes	65
6.11	Quantificação do erro médio para os doze estados nos dois filtros di- ferentes com falhas no DVL e USBL	66

Lista de Acrónimos

AUV	<i>Autonomous Underwater Vehicle</i>
CG	centro de gravidade
DR	<i>Dead-Reckoning</i>
DVL	<i>Doppler Velocity Log</i>
ECI	<i>Earth-Centered-Inercial</i>
EKF	<i>Extended Kalman Filter</i>
ENU	<i>East-North-Up</i>
GAS	<i>Globally Asymptotically Stable</i>
GPS	<i>Global Positioning system</i>
GNSS	<i>Global Navigation Satellite System</i>
IMU	<i>Inertial Measurement Unit</i>
INS	<i>Inertial Navigation System</i>
KF	<i>Kalman filter</i>
LBL	<i>Long baseline</i>
LED	<i>Light-Emitting Diode</i>
MSIS	<i>Mechanical Scanning Imaging Sonar</i>
NED	<i>North-East-Down</i>
PF	<i>Particle Filters</i>
ROS	<i>Robot Operating System</i>
ROV	<i>Remotely Operated Vehicle</i>
SBL	<i>Short baseline</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
USBL	<i>Ultra Short baseline</i>
UKF	<i>Unscented Kalman filter</i>
UUV	<i>Unmanned Underwater Vehicle</i>
VAN	<i>Visually Augmented Navigation</i>

Capítulo 1

Introdução

A superfície do nosso planeta Terra é composta por 71% de água, sendo que os oceanos contemplam quase 97% de toda a água existem. Os oceanos constituem um dos mais importantes recursos naturais para os humanos, apesar destes não serem simples de explorar [10]. Pelo facto da complexidade que é a sua exploração apenas 5 a 7 % dos oceanos foram explorados. Este valor é curto para um meio que constitui um dos mais significativos recursos do qual os seres humanos e animais dependem. Para um melhor estudo e exploração de um recurso tão importante, foram desenvolvidos sistemas e ferramentas como os *Unmanned Underwater Vehicle* (UUV's). Estes são sistemas robóticos que não levam humanos a bordo, sendo ou não controlados por eles.

Consequentemente, a utilização de *Remotely Operated Vehicle* (ROV) (caracterizado por um sistema controlado diretamente pelo homem), pela sua segurança e praticabilidade, provocaram uma enorme repercussão em áreas como a oceanografia e indústria petrolífera. Nada obstante, a necessidade de um veículo de suporte, como um barco, e as restrições na área de trabalho de um ROV, levou à procura e desenvolvimento de novas soluções [11].

Por todas essas razões ao longo dos últimos anos, um grande número de *Autonomous Underwater Vehicle* (AUV's) tem sido desenvolvidos com diferentes aplicações, sejam elas comerciais, militares ou de pesquisa. Estes tendem a ser totalmente autónomos.

Para estudo dos oceanos, os AUV's são utilizados com bastante frequência para a exploração, reconstrução 3D, mapeamento, e para recolha de dados relevantes [12]. Na indústria do petróleo, a inspeção subaquática de infraestruturas, assim como

em tarefas de pesquisa, reconhecimento e localização de missões de recuperação de caixas pretas dos aviões, tem tendência a ser realizados por AUV's [13]. Estes sistemas robóticos tem também como um dos seus possíveis propósitos garantir a segurança de portos, realizando tarefas como vigilância do meio aquático, detecção e desactivação de explosivos e minas [14].

O desenvolvimento deste tipo de robôs encontra desafios diferentes do desenvolvimento de sistemas não subaquáticos, a começar pela comunicação e localização. Fora da água, grande parte dos robôs autónomos emprega sistemas rádio como forma de comunicar, e sistemas, do tipo *Global Positioning System* (GPS), de forma a obter a sua posição global. Dentro de água, o mesmo não é possível devido à fraca propagação de ondas electromagnéticas na água. Tipicamente, para a resolução deste problema, sensores acústicos são adicionados aos AUV's. Devido à complexidade da localização e navegação subaquática, esta é alvo de pesquisa e investimento por parte de muitos centros de investigação, levando ao aparecimento de novas soluções e tecnologias.

De entre os vários métodos existentes, o método *Dead-Reckoning* (DR) é uns dos mais clássicos e por norma a base de um sistema de localização. Neste pode ser utilizado um sistema INS, para obter as acelerações e as velocidades angulares do robô, ou um DVL, para observar as velocidades lineares, sendo que a posição é obtida através da integração dos dados das acelerações, com o INS, ou das velocidades, com o DVL. Para o caso de se querer observar diretamente a posição, um sistema acústico é necessário, sendo estes definidos com base na *baseline* estabelecida; enumerando-os: *Long baseline* (LBL), *Short baseline* (SBL) e USBL.

Quando a necessidade passa por um sistema robusto e preciso, é desenvolvido um algoritmo de fusão sensorial, como por exemplo um *Particle Filters* (PF) ou um *Kalman filter* (KF). Neste tipo de algoritmos, a ideia passa por fundir a informação de vários sensores.

Nesta dissertação será apresentado o desenvolvimento de um sistema de localização, ou seja, um sistema capaz de estimar com precisão os estados relativos à sua configuração num dado referencial, com a maior precisão possível. Integrando vários sensores presentes no AUV.

1.1 Contextualização

O projeto Turtle [15] tinha como finalidade o desenvolvimento de sub-sistemas, para um *lander* robótico, com a capacidade de realizar longas missões em águas profundas. A principal motivação deste era o desenvolvimento de um sistema com baixo consumo energético na manobra de subida e descida dentro de água. Com o passar dos anos e com sucessivos projetos, novas ideias foram surgindo para acrescentar ao

Turtle. Uma das novas alterações é a sua forma de carregamento. Nas versões anteriores para carregar as baterias do *lander* era necessário retirá-lo do mar e depois, fora de água, alimentar as baterias. Todo este processo é complexo, uma vez que implicava solicitar um barco de apoio e deslocar meios e pessoas, de forma a que tudo corresse dentro da normalidade.

A ideia, nesta nova versão, passa pela utilização de um outro AUV para o carregamento do Turtle 3 no mesmo ambiente onde este se insere. Este novo AUV, a partir de agora denominado de sistema/estação de docagem, estará preso, por um cabo, a uma bóia que flutuará à superfície do mar. A bóia possuirá o sistema de carregamento e o cabo transportará a energia da bóia até ao Turtle, sendo que o sistema de docagem fará a ligação entre o cabo de energia e o Turtle. De referir que o cabo pode também transmitir informação entre a bóia e a estação de docagem.

Este sistema de docagem, que já tem definido o seu *hardware*, tem como principal objetivo a docagem no Turtle; permitindo assim o carregamento das baterias. Esta manobra de docagem pode ser dividida em três partes distintas: fase de procura, fase de aproximação e fase de docagem. A fase de procura consiste na fase onde o sistema de docagem e o Turtle se encontram mais afastados, estando este último tendencialmente no fundo do mar, fig. 1.1.

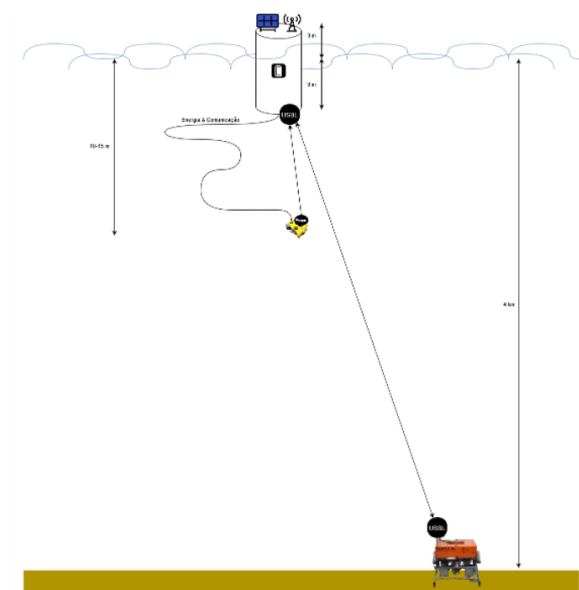


Figura 1.1: Fase de procura

Na fase seguinte, a fase de aproximação, fig. 1.2, o Turtle deslocou-se para o raio de ação do sistema de docagem. Nesta fase será a vez do sistema de docagem deslocar-se de forma a que o Turtle seja visualizado pelo sistema de visão, a bordo do sistema de docagem. Aqui é importante o sistema de docagem possuir um sistema de localização, sendo capaz de se localizar no ambiente de trabalho, e um sistema de localização do sistema de docagem.

↔ Turtle. Este segundo sistema de localização permitirá à doca determinar a zona onde o Turtle está, aproximando assim os dois robôs.

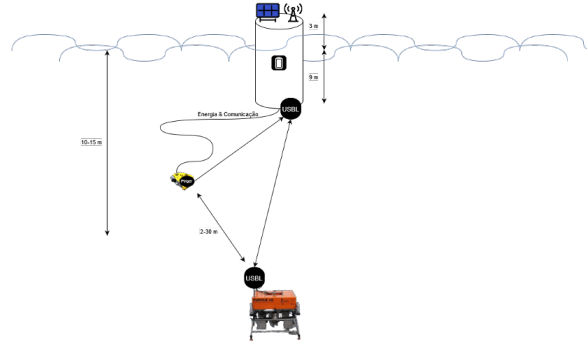


Figura 1.2: Fase de aproximação

A última fase, a fase de docagem, fig. 1.3, caracteriza-se por ser necessário realizar a aproximação de um dos sistemas de modo a se efetuar o encaixe. A câmara, posicionada na parte inferior da doca, torna-se importante, pois recorre-se a esta para a docagem seja realizada de forma suave e que a doca acompanhe possíveis movimentos do Turtle.

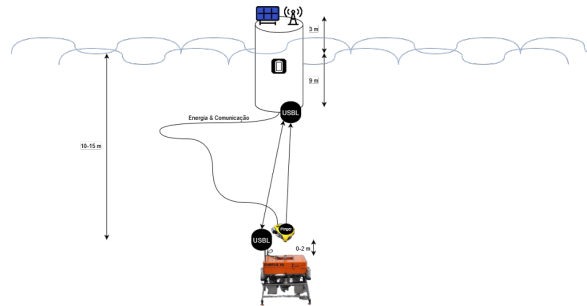


Figura 1.3: Fase de docagem

Todo o processo descrito terá de ser realizado autonomamente obrigado assim ao desenvolvimento de um sistema de localização, permitindo à doca se localizar na água. Para além disso será importante a estação saber onde o Turtle se encontra em relação a ela.

1.2 Motivação e Formulação do problema

A motivação principal desta tese passa pelo desenvolvimento de um sistema de localização que auxilie a doca na manobra de docagem. Para isso será necessário estudar e pesquisar as diferentes abordagens presentes no meio académico.

A doca deverá assim ser capaz de se localizar no seu espaço de trabalho e também capaz de localizar o Turtle em relação a ela mesma. O sistema de docagem estará capacitado com uma série de sensores, tais como, um *beacon* USBL, um sistema INS,

um DVL e uma câmara. O Turtle possuirá um modem acústico USBL, para além dos vários outros sensores não importantes para o desenvolvimento desta tese.

Assim, deverá ser definido uma arquitetura para a solução, com base no *hardware* apresentado, permitindo realizar, com sucesso, a manobra de docagem. Importa referir que os sensores a utilizar no sistema de docagem serão de baixo custo, devido ao baixo custo que a doca deverá ter. Isto terá influência direta na exatidão e precisão da estimação e terá que se ter em conta na hora de desenvolver a estimação.

1.3 Objetivos

Com base no que foi descrito nas secções anteriores é possível escrever uma lista de objetivos capazes de responder ao problema levantado. São estes os seguintes:

- Estudar as diferentes abordagens a sistemas de localização subaquáticos;
- Definição de uma arquitetura do sistema;
- Desenvolver um sistema de localização preciso e exato com sensores de baixo custo;
- Desenvolver um sistema que permita à doca saber a sua localização em relação ao Turtle;
- Validação dos sistemas de localização.

Estudar as diferentes abordagens a sistema de localização subaquáticos permite perceber o estado da arte neste tema e escolher qual a melhor opção a desenvolver. Desse estudo sairá uma arquitetura do sistema com os requisitos apresentados na secção 1.2. A arquitetura terá que contemplar dois sistemas, um que permita a localização da doca no seu referencial de navegação, com base nos sensores do sistema de docagem, e o segundo sistema que possibilite à doca localizar o Turtle, sabendo assim para o onde se terá que deslocar para docar no mesmo. Por fim, será importante validar todo o sistema desenvolvido.

Com os objetivos propostos, a tese terá como *output* um sistema de localização capaz de localizar não só a doca num referencial de navegação mas também localizar o Turtle em relação ao sistema de docagem.

1.4 Organização da Dissertação

A dissertação encontra-se dividida em sete capítulos diferentes. O presente capítulo apresenta a introdução ao documento, assim como a definição do problema e motivação, os objetivos, resultados esperados e o plano de trabalho, seguido ao longo do desenvolvimento do projecto.

No capítulo dois é exposto o estado de arte referente ao desenvolvido. Para além da problemática da localização, é abordado os diferentes métodos de localização subaquática, assim como uma secção onde é explanado a navegação cooperativa. O terceiro capítulo surge de auxílio ao leito, de forma a entender alguns conceitos teóricos utilizados ao longo da dissertação.

A arquitetura da solução final é exposta no capítulo quarto, apresentando o hardware e o software da mesma. Para além disso, é apresentado de forma detalhada os restantes sistemas robóticos integrantes na manobra de docagem.

O capítulo cinco serve de exposição para a implementação da arquitetura apresentada anteriormente; expondo cada bloco dessa mesma arquitetura. O capítulo dos resultados aparece logo a seguir, aqui para além de gráficos a mostrar o desempenho do algoritmo desenvolvido, é apresentado várias tabelas com a quantificação do erro da estimação.

Por fim, as conclusões encontram-se no capítulo sete. Aqui, para além das conclusões a tirar deste projeto, é debatido o trabalho futuro que este poderá ter.

Capítulo 2

Estado da Arte

Este capítulo serve para expor alguns exemplos de sistemas de localização de sistemas subaquáticos encontrados na literatura. Além disso, é abordado em que consiste a localização de um robô, assim como diferentes métodos.

2.1 Problemática da localização

No mundo atual é possível verificar a utilização de sistemas de localização diariamente. Seja nos nossos telemóveis através do sistema GPS [16] ou da triangulação, em automóveis como os da Tesla que usam não só sistema GPS mas também sensores como várias câmaras, para localizar o veículo no espaço [17]. Aliás são sistemas de localização idênticos ao utilizado pela Tesla, com fusão de informação de múltiplos sensores, que vários robôs também utilizam para responder à pergunta "*Onde estou?*".

Assim, um sistema de localização tem como principal objetivo estimar os diferentes estados relacionados com a localização, tais como a posição, a velocidade ou orientação. Para isso é comum equipar os sistemas robóticos com diferentes sensores. Dando o exemplo do sistema GPS, para ambientes *outdoor* a utilização de um sistema de localização com base no GPS é quase mandatário. Por outro lado em ambientes *indoor* ou ambientes aquáticos, a utilização de GPS deixa de ser possível. Serve este exemplo para ilustrar que para diferentes ambientes ou situações é necessário utilizar diferentes sensores, compondo assim diferentes sistemas de perceção.

Ou seja, um sistema de localização tem como objetivo estimar o estado de um robô, sendo que este pode ser obtido através de informação sensorial ou de outros sistemas. Assim para resolver este problema, de onde o robô se encontra, uma técnica utilizada terá que ser capaz de lidar com erros, fatores de degradação, medições inadequadas e estimadores. Estas diferentes técnicas podem ser divididas em duas categorias: localização relativa ou localização absoluta.

Começando pelo sistema de localização relativa, este estima a posição e a orientação através da integração da informação fornecida pela combinação de informação presente em diferentes sensores relativos, normalmente encoders ou sensores inerciais; estes sensores medem informação relativa ao movimento do sistema tal como velocidades, variações de posição ou acelerações. A integração começa na posição inicial e é atualizado continuamente ao longo do tempo. Devido à acumulação contínua de erros no processo de integração, o erro de estimação não é limitado no tempo e por isso tende a crescer. Por essa razão, o *output* deste sistema apenas pode ser utilizado por um período curto de tempo [18].

O sistema de localização absoluta permite a obtenção da localização do robô diretamente do ambiente onde este se insere. Normalmente os métodos associados a este sistema depende de *landmarks* ativas ou passivas, correspondência de mapas ou de sistemas GPS. Ao contrário da localização relativa onde o erro tende a aumentar ao longo do tempo, na absoluta o erro tende a diminuir quando uma medição é recebida. Como a posição do sistema robótico é determinada externamente ao sistema, a sua exatidão não depende do tempo nem da localização. Ou seja, não é necessário a integração de informação, que por sua vez faz com que não ocorra um aumento de erro no tempo ou na distância percorrida. A limitação encontrada nestes tipo de localização é a sua baixa taxa de atualização [18].

No mundo real, os dados obtidos pelos sensores possuem ruído inerente, e os ambientes, onde estes se encontram, são de um certo modo imprevisíveis. Por isso, o processo e resultado do sistema de percepção sofre de um certo grau de incerteza, o que leva à utilização de soluções probabilísticas. Este tipo de solução calcula uma distribuição probabilística sobre onde o robô se pode localizar em vez de apenas gerar a melhor opção de onde ele se encontra.

As abordagens mais clássicas são a utilização de filtros de *Kalman* ou filtro de partículas. Relativamente aos filtros de Kalman, o filtro de Kalman puro apenas pode ser aplicável a modelos lineares. Quando aplicados a modelos não lineares, com incerteza não Gaussiana, novas variantes foram desenvolvidas.

De entre um número de variantes existente, o EKF [19] e o *Unscented Kalman filter* (UKF) [20] são os mais implementados. O EKF utiliza uma abordagem de linearização analítica envolvendo a Jacobiana do modelo do movimento e das medições. Para contornar o problema da limitação da sua aplicabilidade, devido à necessidade de cálculo da Jacobiana, e para aproximar com precisão a não linearidade em termos

de ordem superior, foi desenvolvido o filtro de Kalman *unscented*. Este aproxima os sistemas não lineares utilizando uma transformação *unscented*, o que implica a utilização de um conjunto de amostras chamado pontos *sigma*. Em geral, este tipo de filtros requer mais iterações quando comparado com o EKF.

Em conjunto com os filtros de Kalman, os filtros de partículas [21] são uma das abordagens clássicas, no que toca à fusão sensorial, em localização. Estes utilizam um elevado número de amostras, denominadas de partículas, para representar uma função de distribuição de probabilidade da estimação do estado. Para além disso, funcionam em modelos de movimento não lineares e em movimentos de robôs e medições com incertezas não Gaussianas; não necessitando de obter a Jacobiana. Quando comparando com o EKF, o custo computacional do filtro de partículas pode ser menor que o custo associado ao EKF. No filtro de partículas o custo depende da complexidade do modelo enquanto no EKF depende da complexidade da Jacobiana.

Tal como ficou explícito ao longo do texto, quanto mais sensores se adicionar ao sistema mais facilidade existe em determinar os estados requeridos. Por exemplo, a utilização de um sensor para observar diretamente a velocidade de um robô, torna o sistema mais preciso, quando comparando com a opção de se integrar a aceleração, observada por um *Inertial Measurement Unit* (IMU), para obter a velocidade. Apesar disso, ao se adicionar *hardware* ao sistema, outro problema aparece. A calibração torna-se uma peça importante de um sistema de localização, de forma a tentar eliminar erros provocados pelo desalinhamento ou interferências. Assim algumas das decisões a tomar em relação aos sistemas de localização são [11]:

- Orçamento disponível interfere com a qualidade dos sensores (precisão e exatidão);
- O ambiente de trabalho do robô importa para o tipo de sensores;
- Modelos probabilísticos versus o custo computacional associados a estes;
- O número dos sensores em comparação com a complexidade que esse mesmo número.

2.2 Métodos de localização subaquático

Tal como ficou explícito anteriormente, diferentes sensores promovem diferentes formas de localização subaquática. No meio ambiente onde o mundo se comporta de forma diferente, é desafiante desenvolver novas formas de localizar um sistema. Por toda a complexidade inerente à localização subaquática, é que apenas 5 a 7 % dos oceanos foram examinados.

Tipicamente, a localização subaquática pode ser catalogada em três técnicas diferentes: navegação inercial ou *Dead-Reckoning*, navegação geofísica e navegação

acústica. Sendo o indicado a utilização de um sistema que contemplem diferentes sensores dos apresentados acima, criando assim uma redundância benéfica ao AUV [22]. A figura 2.1 apresenta um esquema de como se divide e que tipo de sensores é utilizado estas três técnicas [1].

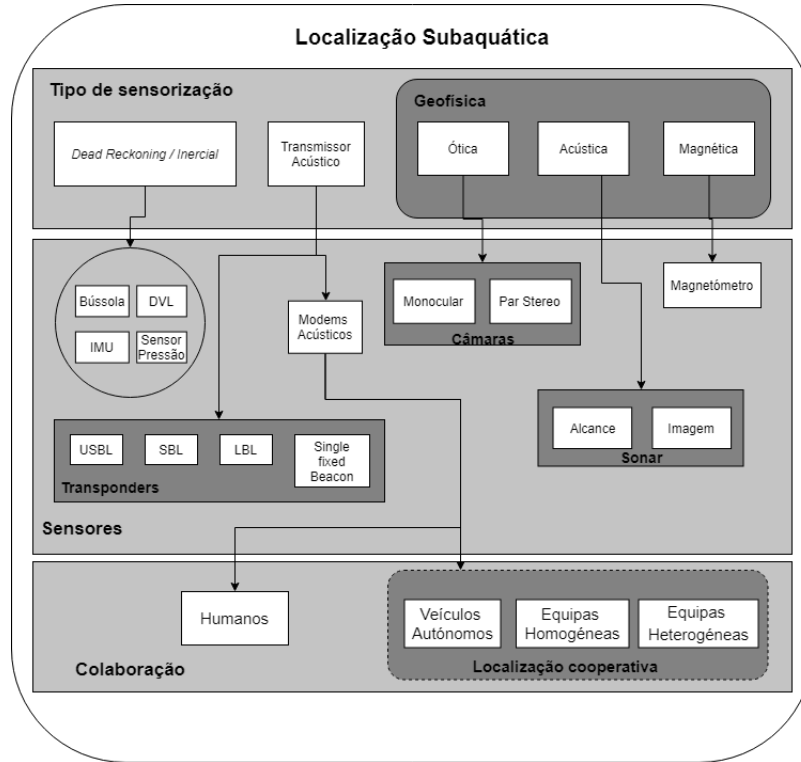


Figura 2.1: Várias técnicas para localização subaquática, baseado em [1]

2.2.1 *Dead-Reckoning* e Inercial

Começando pela navegação inercial ou *Dead-Reckoning*, este método é, dos três apresentados, o mais simples de determinar a posição de um robô, uma vez que apenas é necessário integrar a velocidade em relação ao tempo. Assim com recurso a uma bússola e um sensor de velocidade da água, é possível conhecer a velocidade e a direção do veículo e como consequência, determinar a sua posição ao longo do tempo.

A simplicidade associada a este método leva a que existam desvantagens na sua utilização. Uma destas encontra-se relacionado com alguns fatores externos ao sistema, como por exemplo as correntes marítimas. A possível existência de correntes marítimas leva ao aparecimento de uma componente lateral no vetor de velocidade do AUV. Esta nova componente não é detetada nem pela bússola nem pelo sensor de velocidade. Assim a exatidão deste método fica comprometida [2].

Para contornar esta desvantagem são utilizados sensores inerciais, tipicamente esta é abordagem utilizada no que toca à navegação *Dead-Reckoning*; sensores como INS e DVL. No caso do INS consiste num sistema composto por um giroscópio e três acelerómetros ortogonais alinhais com o referencial do giroscópio. As acelerações medidas sofrem uma integração dupla, de forma a obter a posição do veículo ao longo do tempo. A orientação do sistema é obtida através da integração dupla das acelerações angulares.

A utilização de um sistema INS apresenta algumas desvantagens. Os acelerómetros que compõe o INS estão sujeitos a *drift* ao longo do tempo. O que implica que, quando é utilizado apenas um INS, exista uma acumulação gradual do erro em posição ao longo do tempo. A causa deste erro é a diferença entre a integração da aceleração e o *output* do giroscópio. Para além disso, a incerteza nas medições pode ser causada por forças externas como as correntes marítimas; que podem facilmente alterar a posição do AUV e afetar a performance do INS. Ademais este tipo de eventos é difícil de medir com sensores inerciais.

Para reduzir estes erros de *drift*, especialmente em missões longas, Stutters apresenta em [23] uma abordagem onde utiliza um DVL para a medição da velocidade do fundo oceânico em relação ao AUV. Portanto, fundindo os dados do DVL com o INS reduz o *drift* do veículo ao longo do tempo. Não obstante, este tipo de sensores, DVL, apresentam um alcance limitado e por isso, o AUV necessita de se encontrar perto do fundo do mar [23].

Uma outra solução para a redução do *drift* é apresentado no projeto de Panish e Taylor [24]. Neste, a equipa com recurso a um giroscópio laser, um giroscópio ótico e um DVL de alta qualidade com sensores inerciais. Como apresentado nos resultados fica provado que utilizar um IMU de alta qualidade limita o *drift* e fornece uma performance excelente para a navegação do AUV [24].

2.2.2 Geofísica

A navegação geofísica caracteriza-se por ser a forma de navegação que faz uso das características físicas do meio ambiente para estimar a localização do AUV. Para se obter o descrito é necessário a utilização de um sonar, um sensor ótico ou um magnetómetro. Estes sensores vem por norma sendo aplicados na deteção, identificação e classificação das características físicas do ambiente.

Contudo, devido à baixa resolução destes e às diferentes formas geométricas irregulares das *features* oceânicas, é difícil conseguir extrair as características do meio através do sensor. De forma a mitigar este problema, câmaras subaquáticas podem ser utilizadas, uma vez que são considerados sensores como maior precisão. Apesar disso, este tipo de sensores apresentam as suas desvantagens, tais como o seu alcance. Por isso, para utilização de uma câmara subaquática é necessário ao

veículo encontrar-se perto do objeto de interesse [11] ou utilizar algo externo com luz de forma a iluminar o meio.

Os dois tipos de mapas subaquáticos que podem ser utilizados como referência para informação geofísica são os mapas de sonar batimétrico e campo magnético. Portanto, os AUV's podem utilizar estes mapas da área para navegar ou se o mapa não se encontrar disponível, os veículos podem construir o mapa ao longo da missão, ajudando assim ao sistema a navegar no meio até então desconhecido. As posições das características externas são fixas e conhecidas à partida. Assim, quando o veículo deteta e identifica uma dessas características, utiliza a posição conhecida dessa *feature* como ponto de referência, corrigindo os erros de navegação.

Uma das técnicas com recurso a métodos geofísico mais utilizadas na navegação de sistemas autónomos, é denominada de *Simultaneous Localization and Mapping* (SLAM). Esta caracteriza-se pelo processo onde o robô consegue autonomamente construir um mapa do seu meio de trabalho e ao mesmo tempo utilizar esse para se localizar e navegar. Este método consiste em dois procedimentos: mapeamento e sensorização. Para o mapeamento é gerado uma estimação de um mapa local através da extração de *landmarks* do meio ambiente, e o procedimento de sensorização é a estimação do posicionamento do robô num mapa local gerado pela procedimento do mapeamento [2].

A título de exemplo, no artigo [25] é apresentada uma solução onde as paredes são a principal fonte no processo de localização. Estas são identificadas através de um *Mechanical Scanning Imaging Sonar* (MSIS). Um *priori-map* é utilizado e com a transformada de Hough é possível detetar as retas que formam as paredes. Por fim, um filtro EKF é empregue para a estimação dos estados.

No paper de Eustice [26] foi aplicado um *Visually Augmented Navigation* (VAN) de forma a melhorar a exatidão da navegação de um AUV baseada em SLAM. O processo é realizado com recurso a uma câmara, de forma a obter imagens do fundo do mar; com a sequência de imagens obtidas, obtém-se a velocidade e a atitude do AUV. Importa referir que se volta a recorrer a um filtro EKF.

2.2.3 Acústica

A navegação acústica é uma das técnicas de navegação subaquática mais utilizada, verificando-se na variedade de sistemas existentes com base neste tipo de navegação. Para esta navegação subaquática não basta a colocação de um sensor no AUV, é necessário a utilização de uma *baseline* de transmissores acústicos, capazes de comunicar com o sensor colocado no robô. A caracterização deste tipo de sensores acústicos depende da distância entre *acoustic baselines*. Três tipos de sistemas são, normalmente, utilizados: USBL, com distâncias inferiores a 10 m, SBL, entre 20 e 50 m e LBL com distâncias entre 100 e 6000 m [27].

A posição estimada do sistema autônomo é relativa a um *local reference frame* e pode ser calculada com base nas distâncias entre cada *transducer*, colocados numa *baseline*, e o *transponder*, presente no AUV.

Um sinal acústico é enviado pelo *transceiver*, que contém numa *baseline* vários *transducers*, e detetado pelo *transponder*. Este responde com um novo sinal acústico que é detetado pelo *transponder*. O intervalo de tempo entre a transmissão do sinal acústico inicial e a deteção da resposta, por parte do *transceiver*, é medido e utilizado para calcular a distância. Para o cálculo do ângulo entre o *transceiver* e o *transponder* é utilizado a diferença de tempo entre a receção do sinal por cada *transducers*. Ora como estes se encontram numa *baseline* separados por uma certa distância, o intervalo de tempo calculado anteriormente vai ser diferente para cada *transducer*, essa diferença é fundamental para a determinação do ângulo.

O sistema de localização acústica utiliza sinais acústicos uma vez que estes apresentam pouca absorção na água quando comparando com os sinais eletromagnéticos. Traduzindo assim uma das vantagens da utilização de sinais acústicos, uma vez que estes percorrem maior distâncias na água do que os sinais eletromagnéticos. Os sensores mais comuns da navegação acústica são LBL, que utiliza dois *beacons* colocados no fundo do mar ou à superfície e separados por uma distância relativamente grande. O sistema SBL caracteriza-se por três *beacons* distintos e separados por uma certa distância, instalados no casco de uma bóia, por fim, o USBL que apenas utiliza um *beacon* colocado na parte inferior da bóia flutuante [2]. Na figura 2.2 é apresentado os três sistemas distintos abordado anteriormente.

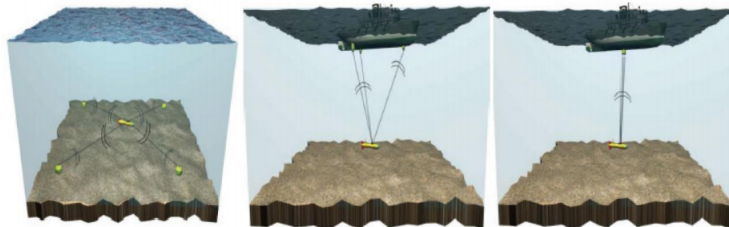


Figura 2.2: Descrição gráfica de cada um dos sistemas: LBL, SBL e USBL [2]

Em todos os três tipos de sistemas, o *transponder* encontra-se instalado no AUV, enquanto os *transducers* são colocados em sítios diferentes, dependendo do sistema a utilizar. Adicionalmente, e uma vez que a localização de todos os *transducers* é conhecida, é possível criar uma *local reference frame* e seleccionar a localização do ponto de origem e as direções dos eixos horizontais e verticais. Por isso, com o conhecimento da posição dos *beacons*, do tempo de voo do sinal, e da velocidade deste, o AUV consegue calcular a sua localização através da intersecção das posições relativas do AUV em relação a cada um dos *beacons*.

A principal diferença entre os três métodos é a distância da *baseline* e a localização dos *transducers*. A discussão acerca da localização foi discutida anteriormente. Em relação à distância da *baseline*, o sistema USBL apresenta a mais pequena das *baselines*, sendo inferior a 10 cm, enquanto o sistema LBL apresenta a maior *baseline* com um alcance entre 100 e 6000 metros. No sistema SBL, o comprimento da *baseline* é entre 20 e 50 metros, o que é adequado para instalar a mesma no casco do AUV.

Cada um dos sistemas de posicionamento acústico apresenta vantagens dependendo do tipo de missão. Por exemplo, um sistema de posicionamento SBL é melhor para uma missão de *tracking* e de um alcance curto, enquanto para um sistema LBL é utilizado para missões de maior alcance. Uma das vantagens de todos estes sistemas de acústicos é que estes providenciam uma localização absoluta do AUV, relativa à *frame* da *baseline* acústica. O mesmo não acontece quando se utiliza um INS para a obtenção dessa informação. Ou seja, esta vantagem faz com que o sistema de posicionamento acústico funcione de forma idêntica ao sistema de GPS, mas dentro do meio aquático.

Este tipo de modo de navegar no oceano tem sido usado com bastante frequência em sistemas autónomos; em 2008, um grupo de investigação fez uso de um sistema acústico LBL de forma a fornecer a posição a um AUV para operar por cima de uma camada de gelo. Neste exemplo de utilização, foi colocado, no fundo do oceano, dois *beacons*, parados e espaçados, e vários *beacons* presos ao AUV. Assim, a posição do veículo é calculada medindo o tempo de voo dos sinais, do AUV até cada um dos *beacons*.

Tipicamente, para melhorar a eficácia do sistema acústica é necessário combinar este com outros sistemas, como por exemplo com INS. No exemplo, Ridao [28] a informação proveniente de um sistema USBL é fundida, com recurso a um filtro, com os dados de um INS; reduzindo assim os erros. O USBL fornece a posição absoluta que ajudam a minimizar os erros de *drift*, comuns na utilização de um INS.

Contudo, existe algumas desvantagens neste tipo de navegação. Na abordagem LBL, os sistemas necessitam de ser instalados por longos períodos de tempo, o que leva a um aumento do custo associado à missão. Por outro lado, SBL utiliza uma técnica onde o sistema necessita de ser instalado no veículo. Isto faz aumentar a relação sinal-ruído e degrada a exatidão do sistema. A grande desvantagem associada à utilização do USBL encontra-se no facto deste tipo de sistema apenas ser exato em ambientes com águas calmas.

2.2.4 Estimação e Filtragem

A maioria dos AUV atuais possuem vários sensores que são utilizados para melhorar a estimação da *pose* do veículo. Cada sensor fornece, ao sistema, informação que necessita de ser fundida com informação de outros sensores ao longo da missão; assim

obtem-se uma estimaco da posico do AUV otimizada. Este processo   chamado de filtragem. Paull, Saeedi, Seto, and Li [1] trabalharam com m todos diferentes de estimaco de estados utilizados na navegaco de AUV.

No paper de Yun [29]   utilizado um filtro complementar de 12 vari veis de estado para fundir dados de sensores como um m dulo GPS, um IMU, uma b ssola, um sensor para determinar a velocidade da  gua e um sensor de press o. Os resultados, atrav s da simulaco, fornecem uma abordagem pr tica para calibrar os ganhos do filtro. Para al m disso, atrav s do uso de um *ground vehicle*   verificado o funcionamento do filtro complementar, assim como o seu grau de exatid o.

No projeto de Wang [30]   comparado dois filtros diferentes, o EKF e o UKF. A conclus o do paper   que a utilizaco de um UKF, para a estimaco dos estados do AUV,   mais exato e reduz mais o erro, quando comparando com o EKF. Isto acontece devido a uma grande n o-linearidade nos seus modelos dos ve culos aqu ticos, e por isso, o EKF t pico n o funciona de forma eficaz nessa situaco. Ademais, existem v rios papers [31] e [32] que tamb m concluem que o UKF tende a obter melhores resultados que o EKF.

Ali s, Rigby, Pizarro, e Williams [33] utilizam um m todo de fus o de dados de multi-sensores de forma a combinar informaco de sensores como USBL, DVL e um girosc pio. Em vez de utilizar um filtro conhecido, como os abordados anteriormente, eles desenvolveram um algoritmo de filtragem como o nome de *Globally Asymptotically Stable* (GAS); de forma a fundir as medi  es n o lineares dos tr s sensores. Os resultados da simulaco mostram que o filtro GAS foi capaz de obter resultados id nticos ao EKF.

2.3 Navegaco Cooperativa

Neste tipo de navegaco subaqu tica, equipas de AUV utilizam n o s  os sensores de cada ve culo como comunicaco entre os v rios membros da equipa.

No paper escrito por Roumeliotis e Bekey em [34]   provado que um grupo composto por AUV's, sem acesso a um posicionamento global, podem se localizar a eles mesmo com precis o se partilharem a sua posico estimada e a incerteza. Aplica  es direccionadas   navegaco cooperativa com recursos a um EKF [35], ao algoritmo dos m nimos quadrados [36], e estimaco m xima   posteriori [37] tem sido desenvolvidas e estudadas.

Uma equipa heterog nea caracteriza-se por um dos membros desta possuir sensores de maior capacidade ou por um dos membros realizar viagens   superf cie, para corrigir a posico com base no GPS, mais frequente. Um exemplo disto seria existir uma frota de AUV's com sensores de baixo custo e um n mero reduzido de ve culos com sensores de alta gama, que suportariam essa mesma frota [38]. Com a receco de uma mensagem de navegaco   poss vel restringir a estimaco da posico

do recetor a um anel, se cada veículo tiver uma boa estimaco da profundidade (por exemplo, com um sensor de presso) e se o recetor conseguir estimar a distncia para o transmissor (por exemplo, atravs da medio do tempo de voo de um sinal acstico). Uma aplicao deste tipo de equipas  apresentado em [39], onde um AUV do estilo flutuador consegue realizar a docagem num AUV do estilo torpedo de forma a trocar informao.

Noutro tipo de equipas de AUV's, uma equipa homognea caracteriza-se por tratar cada membro da equipa de forma igual. O erro em posio crescer mais lento se o AUV's so capazes de comunicar a sua posio relativa e distncias. Semelhante ao caso das equipas heterogneas, caso algum vculo vem  superfcie, de forma a corrigir a sua posio, ento a informao pode ser partilhada com todo a equipa de forma a limitar o erro da posio.

A primeira experincia conhecida da implementao deste tipo de sistema foi em 2007 [40]. Nesta abordagem, os estados dos sensores so obtidos atravs de um filtro EKF mas as covarincias no so partilhadas devido  largura de banda da comunicao. A latncia da comunicao  tida em conta no filtro. Um outro exemplo deste tipo de abordagem  apresentado em [41]. Neste  criado uma espcie de registo multiveculo de forma a que a atualizao dos estados previstos no EKF apenas seja realizado quando toda a informao tenha sido recolhida. A experincia foi realizada em caiaques, simulando AUV's, onde a determinao da distncia, atravs do tempo de voo do sinal, foi realizada com base em relgios sincronizados num ciclo de 30 segundos, tendo, por isso, cada um dos trs vculos um tempo de 10 segundos para realizar a transmisso.

Capítulo 3

Conceitos Teóricos

Neste terceiro capítulo é apresentado alguns conceitos teóricos, utilizados ao longo do desenvolvimento do projeto, tais como o conceito de referencial, as diferentes formas de representar uma orientação em 3D, as transformações de referencial e o filtro de Kalman estendido. Para além disso, é abordado, de forma teórica, o modelo hidrodinâmico a ser utilizado neste projeto.

3.1 Referenciais

Um sistema de localização caracteriza-se por um sistema que tenha como output a informação acerca do estado de um veículo. Este estado, seja posição ou orientação ou ambos, é relativos a um referencial. Existem vários referenciais pré-definidos, sendo estes úteis pois permite relacionar a informação proveniente de cada sensor; permitindo assim, facilmente, estimar a *pose* de um veículo com base nesses sistemas ou sensores.

3.1.1 Referencial Inercial (*i-frame*)

Os referenciais inerciais tem como sua grande aplicação os sensores inerciais, pois estes tem a possibilidade de fornecer dados nesse tipo de referencial. Estes referenciais são, em norma, definidos como um referencial para o qual se realiza o movimento linear [3].

Um exemplo de um referencial inercial é o *Earth-Centered-Inercial* (ECI) que apresenta as seguintes características:

- O origem das coordenadas é o centro de gravidade da Terra;
- Os três eixos referenciais apontaram para:
 - O eixo X para o equinócio vernal;
 - Paralelo ao eixo de rotação da Terra;
 - Definido pela regra da mão direita do sistema de coordenadas ortogonais.

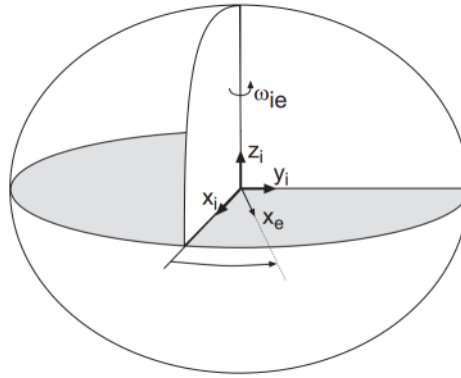


Figura 3.1: Referencial inercial [3]

3.1.2 Referencial Local (*n-frame*)

Um referencial local consiste numa superfície plana tangencial num ponto na Terra onde a navegação ocorre. Dependendo das aplicações existem dois referenciais locais mais utilizados, são eles o referencial *North-East-Down* (NED) e o *East-North-Up* (ENU). O NED é utilizado em veículos subaquáticos e aéreos, enquanto nos veículos terrestres a escolha recai sobre o ENU. A origem de ambos os sistemas é arbitrária, apesar de que se devem encontrar fixos ao veículo, movendo-se assim com o referencial do corpo [11].

No caso do referencial NED, este é definido de tal forma que os eixos Norte e Este formam um plano tangente à superfície da Terra, assumindo um modelo elipsóide WGS84 da Terra. Tal como mostrado na figura 3.2, o referencial contém três eixos ortogonais em que o eixo N_x aponta para o Norte verdadeiro, o N_z para o interior da Terra e o N_y completa o sistema apontado para Este.

Idêntico ao referencial NED, existe o referencial ENU que, tal como o NED, pode ser colocado localmente no veículo, movendo-se com ele. Este difere no NED na direção dos seus eixos ortogonais. O eixo N_y aponta para o Norte verdadeiro, o N_z aponta para o exterior do centro da Terra e por fim, o eixo N_x aponta para Este [42].

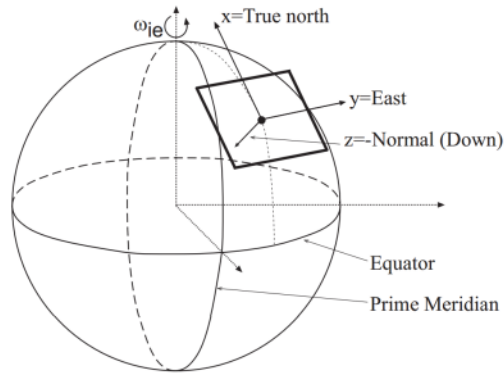


Figura 3.2: Referencial Local NED em relação ao referencial ECEF [4]

3.1.3 Referencial do Corpo (*b-frame*)

Na maior parte das aplicações, a obtenção da posição, velocidade ou orientação é realizada com ajuda de sensores fixos ao veículo. Este veículo pode ter o seu próprio referencial, denominado referencial do corpo. Estes caracterizam-se por referenciais onde a origem deste é colocado no centro de gravidade do veículo e três eixos ortogonais que respeitem a regra da mão direita. Esta orientação pode ser expressa em ângulos de Euler em torno dos eixos x^b, y^b, z^b [42].

Na navegação aquática, o movimento sobre o eixo X é denominado por *surge*, sobre o eixo Y por *sway* e sobre o eixo vertical, o Z , por *heave*.

Importa referir que em aplicações onde o objetivo passa por estimar a *pose* de um veículo, é necessário definir para cada sensor e para o veículo um referencial para cada um desses "corpos". Quando, em alguns exemplos, não é possível que o referencial do sensor se encontre perfeitamente alinhado com o referencial do veículo, é necessário realizar uma rotação para alinhar os dois referenciais [42].

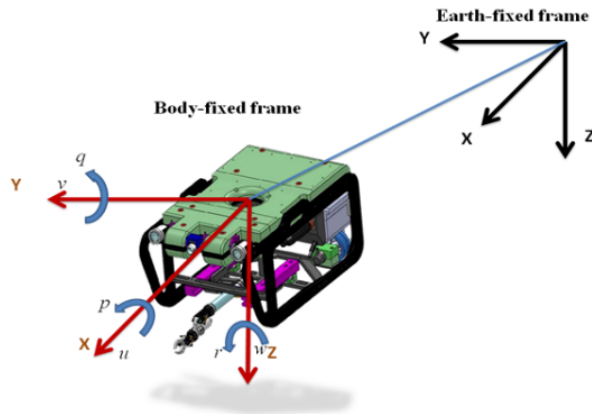


Figura 3.3: Exemplo de um referencial do Corpo (*body frame*) [5]

3.2 Rotações tridimensionais

Das diferentes formas de representação de uma orientação de um corpo em três dimensões, os ângulos de Euler e os quaterniões são os mais utilizados.

3.2.1 Ângulos de Euler

Pela razão de ser de mais fácil compreensão ao ser humano, os ângulos de Euler é o método mais utilizado; O nome deve-se ao matemático que os formulou, Leonhard Euler [43].

A orientação é descrita a partir de três ângulos, associados a cada um dos eixos de um plano cartesiano, x , y e z . Tradicionalmente, os ângulos são denominados de *roll* (ϕ), rotação em torno do eixo X , *pitch* (θ), rotação em torno do eixo Y , e *yaw* (ψ), em torno do eixo Z .

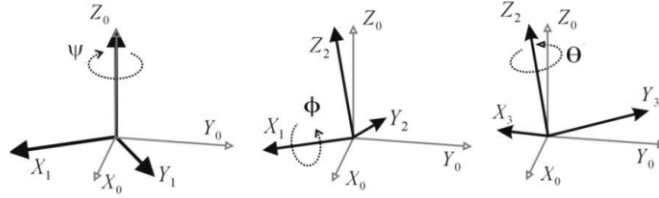


Figura 3.4: Representação dos ângulos de Euler [6]

Tratando a questão de forma matemática, a rotação em torno do eixo X (ϕ) pode ser exposta através da seguinte matriz:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (3.1)$$

A rotação em torno do eixo Y é representada matematicamente com a seguinte matriz:

$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (3.2)$$

Por fim, a matriz que representa a rotação em torno do eixo Z é a seguinte:

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Ao aplicar as três rotações a cada um dos eixos, obtemos a seguinte matriz de rotação:

$$R(\phi, \theta, \psi) = R_x(\phi)^T R_y(\theta)^T R_z(\psi)^T \quad (3.4)$$

$$= \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \sin(\phi)\cos(\theta) \\ \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) & \cos(\phi)\cos(\theta) \end{bmatrix} \quad (3.5)$$

Com base na equação 3.5 é possível obter os ângulos de Euler com a seguinte equação:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(R_{23}, R_{33}) \\ -\text{asin}(R_{13}) \\ \text{atan2}(R_{12}, R_{11}) \end{bmatrix} \quad (3.6)$$

3.2.2 Quaterniões

Uma outra forma de representar uma orientação de um objeto em três dimensões é utilizando quaterniões. Estes, contrariamente aos ângulos de Euler, não são de fácil compreensão para o ser humano e por essa razão tendem a não ser utilizados. Apesar disso, os quaterniões resolvem um problema associado aos ângulos de Euler, as singularidades ou *gimballocks* [44].

Dando um exemplo, com um *gimbal*, daí o nome *gimballocks*, uma singularidade ocorre quando, na representação *roll-pitch-yaw*, o ângulo de *pitch* é igual a $\pi/2$. Assim, os eixo de *yaw* e de *roll* são paralelos, levando a que a rotação em *yaw* possa não ser compensada [44].

Os quaterniões são representados na seguinte forma:

$$q = [q_0 \quad q_1 \quad q_2 \quad q_3]^T \quad (3.7)$$

onde q_0 é um escalar e os restantes, q_1 , q_2 e q_3 são números complexos.

Para transformar ângulos de Euler para quaterniões, a relação é a seguinte:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\frac{\phi}{2})\cos(\frac{\theta}{2})\cos(\frac{\psi}{2}) + \sin(\frac{\phi}{2})\sin(\frac{\theta}{2})\sin(\frac{\psi}{2}) \\ \sin(\frac{\phi}{2})\cos(\frac{\theta}{2})\cos(\frac{\psi}{2}) - \cos(\frac{\phi}{2})\sin(\frac{\theta}{2})\sin(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2})\sin(\frac{\theta}{2})\cos(\frac{\psi}{2}) + \sin(\frac{\phi}{2})\cos(\frac{\theta}{2})\sin(\frac{\psi}{2}) \\ \cos(\frac{\phi}{2})\cos(\frac{\theta}{2})\sin(\frac{\psi}{2}) - \sin(\frac{\phi}{2})\sin(\frac{\theta}{2})\cos(\frac{\psi}{2}) \end{bmatrix} \quad (3.8)$$

Assim, e com base nas equações 3.5 e 3.8, obtém-se a matriz de rotação em quaterniões:

$$R = \begin{bmatrix} -q_3^2 - q_2^2 + q_1^2 + q_0^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_0q_3 - q_1q_2) & -q_3^2 + q_2^2 - q_1^2 + q_0^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_3^2 - q_2^2 - q_1^2 + q_0^2 \end{bmatrix} \quad (3.9)$$

3.3 Transformação de Referenciais

As transformações de referencias conhecidos permitem representar informação em diferentes sistemas de coordenadas. Isto torna-se particularmente importante pois num sistema de navegação é essencial que informação dos diferentes sensores se encontre no mesmo referencial. Abaixo é apresentado diferentes transformações utilizadas no decorrer deste projeto.

3.3.1 Transformação *b-frame* para *n-frame*

Para a transformação entre o referencial do corpo e o referencial de navegação é necessário encontrar a rotação que os relaciona. Assim, a relação entre estes dois referenciais é dado por:

$$v_n = R_b^n v^b \quad (3.10)$$

onde v_n e v_b são as coordenadas no referencial de navegação e no referencial do corpo, respetivamente, e R_b^n a matriz de rotação transposta da equação 3.5 e expressa da seguinte forma:

$$R_b^n = \begin{bmatrix} \cos(\theta)\cos(\psi) & -\cos(\phi)\sin(\psi)+\sin(\phi)\sin(\theta)\cos(\psi) & \sin(\phi)\sin(\psi)+\cos(\phi)\sin(\theta)\cos(\psi) \\ \cos(\theta)\sin(\psi) & \cos(\phi)\cos(\psi)+\sin(\phi)\sin(\theta)\sin(\psi) & -\sin(\phi)\cos(\psi)+\cos(\phi)\sin(\theta)\sin(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \quad (3.11)$$

3.3.2 Transformação entre Referencial Local ENU e NED

Para transformar um vetor de coordenadas do referencial local ENU para o referencial local NED é necessário aplicar duas rotações. Uma de π rad sobre o eixo X do referencial ENU fazendo com que o eixo Z em ENU fica alinhado com o eixo Z do sistema NED. De seguida, aplicar uma rotação $-\frac{\pi}{2}$ rad sobre o novo eixo Z de forma a alinhar os eixos X e Y com o novo referencial local NED.

Concluindo, a matriz de rotação que descreve a transformação ENU para NED é dada por:

$$R_E^N = \begin{bmatrix} \cos(-\frac{\pi}{2}) & -\sin(-\frac{\pi}{2}) & 0 \\ \sin(-\frac{\pi}{2}) & \cos(-\frac{\pi}{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\pi) & -\sin(\pi) \\ 0 & \sin(\pi) & \cos(\pi) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.12)$$

Com esta matriz de rotação, 3.12, é possível verificar que para transformar as coordenadas ENU para NED, basta trocar a coordenada X e Y entre si, e multiplicar a coordenada Z por -1 [45].

3.4 EKF

O filtro EKF é um algoritmo de estimação utilizado em várias áreas envolvendo estimação de estados de sistemas dinâmicos, a partir de medições de sensores. Quando considerando um sistema dinâmica não linear, a escolha recai nesta versão do filtro de Kalman. Tal como referido anteriormente, a informação recolhida de sensores é fundamental para a implementação de um EKF, sendo que é esta que é combinada com as previsões do estado atual obtido [46].

O algoritmo divide-se em dois momentos, a previsão e a atualização ou *update*.

A fase da previsão, e tal como o nome indica, serve para se prever o estado estimado e a sua covariância associada. Nesta fase é utilizado a estimativa anterior (estado e covariância), o modelo dinâmico do sistema e as entradas do sistemas entre o instante da última estimação e o instante atual. Estes são obtidos através das seguintes equações:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}) \quad (3.13)$$

onde $\hat{x}_{k|k-1}$ é o estado a prever, f a função que representa o modelo de previsão (modelo da dinâmica do sistema) e u_{k-1} as entradas do modelo de previsão, que normalmente são controlos ou medidas de sensores inerciais.

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T \quad (3.14)$$

onde $P_{k|k-1}$ a covariância do estado prevista, a F_{k-1} a matriz jacobiana do modelo de previsão em ordem às variáveis de estado, G_{k-1} a matriz jacobiana do modelo do sistema em ordem as fontes de ruído e por fim, Q_{k-1} a covariância do ruído.

Após a fase de previsão, é a fase de atualização. Nesta os estados e respetivas covariâncias prevista, obtidos na fase anterior, são atualizados de acordo com as observações. Estas observações são as medições dos sensores no instante de tempo que se faz a atualização.

Aqui para além de ser necessário calcular o estado estimado e a sua covariância, é importante alguns cálculos auxiliares, tais como o cálculo da inovação, a covariância da inovação e o ganho de Kalman.

A inovação consiste na diferença entre as observações previstas z_k , utilizando o modelo de observação, e a melhor previsão do estado do sistema. O modelo de

observação é representado por h , e relaciona os estados previstos com as observação previstas, sendo H_k o jacobiano dessa função em ordem às variáveis de estado.

$$\bar{y}_k = z_k - h(\hat{x}_{k|k-1}) \quad (3.15)$$

A covariância associada à inovação aponta o nível de incerteza relativa à própria inovação, onde R_k é a matriz de ruído da medida.

$$S_k = H_k P_{k|k-1} (H_k)^T + R_k \quad (3.16)$$

De seguida, o Ganho de Kalman. Este indica o peso de aceitação do sistema relativo à nova observação.

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (3.17)$$

Para o cálculo do estado estimado, tem-se por base o estado previsto somando-se a multiplicação do ganho de Kalman com a inovação

$$\hat{x}_{k|k} = \hat{x}_{k-1|k-1} + K_k \bar{y}_k \quad (3.18)$$

Por fim, a covariância estimada, que é a incerteza do novo estado estimado é calculado da seguinte forma:

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \quad (3.19)$$

3.5 Modelo hidrodinâmico

Nesta secção é apresentado o modelo hidrodinâmico utilizado para o desenvolvimento deste projeto. Este é obtido diretamente do livro *Handbook of marine craft hydrodynamics and motion control* de Thor Fossen[47].

Na tabela 3.1 é apresentada e descrita a convenção dos estados de um veículo aquático.

Tabela 3.1: Variáveis associadas ao estado de um veículo aquático

Termo	Posição	Velocidade	Aceleração	Força
Surge	x	u	$\dot{u}$	X
Sway	y	v	$\dot{v}$	Y
Heave	z	w	$\dot{w}$	Z
Roll	ϕ	p	$\dot{p}$	K
Pitch	θ	q	$\dot{q}$	M
Yaw	ψ	r	$\dot{r}$	N

As forças que atuam num veículo subaquático podem ser divididas em cinco categorias: Forças cinemáticas, hidrodinâmicas, hidrostáticas, forças dos atuadores

e perturbações. Assumindo um modelo de um veículo subaquático com seis graus de liberdade, este pode ser representado pelas seguintes equações [47]:

$$\dot{\eta} = J(\eta)v \quad (3.20)$$

$$M\dot{v} + C(v)v + D(v)v + g(\eta) + g_0 = \tau + \tau_{wind} + \tau_{wave} \quad (3.21)$$

Na lista abaixo é explanado os termos utilizados nas equações acima:

η - Posição e Pose do veículo

v - Velocidade do veículo na *b-frame*

J - Matriz de transformação

M - Matriz inercial

$C(v)$ - Forças de Coriolis

$D(v)$ - Matriz de *Damping*

$g(\eta)$ - Força de *Buoyancy*

g_0 - Força devido ao lastro

τ - Força dos atuadores

τ_{wind} - Efeito do vento

τ_{wave} - Efeito das ondas

O modelo dinâmico consiste nos termos e nos coeficientes apresentados na tabela 3.2.

De forma a obter o modelo dinâmico, primeiramente é modelado as forças dos atuadores de forma separada e só depois substituídas na equação 3.21. O veículo é considerado um corpo rígido e por isso não é expectável que sofra nenhuma deformação ao longo da sua missão. Com base nesta afirmação a cinemática de um corpo rígido (τ_{RB}) pode ser expressa pela soma da sua inércia ($M_{RB}\dot{v}$) e os efeitos de Coriolis ($C_{RB}(v)v$), tal como demonstrado na equação 3.22 [47].

$$\tau_{RB} = M_{RB}\dot{v} + C_{RB}(v)v \quad (3.22)$$

Para um modelo de seis graus de liberdade, e assumindo que o centro de origem e o centro de gravidade (CG) não coincidem, M_{RB} é representado da seguinte forma:

Tabela 3.2: Lista dos termos e coeficientes do modelo dinâmico

Coeficiente	Significado
m	Massa do veículo
W	Peso do veículo
B	Força de <i>Bouyancy</i>
g	Aceleração da gravidade
ρ	Densidade da água
I	Momento de inércia
I_x, I_y, I_z	Momentos de inércia em torno do eixo do x, y e z respectivamente
$X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}$	Massa adicionada devido à translação
$K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}$	Inércia adicionada devido à rotação
X_u, Y_v, Z_w	Coefficientes Linear Damping devido à translação
$X_{u u}, Y_{v v}, Z_{w w}$	Coefficientes Quadratic Damping devido à translação
K_p, M_q, N_r	Coefficientes Linear Damping devido à rotação
$K_{p p}, M_{q q}, N_{r r}$	Coefficientes Quadritc Damping devido à rotação

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_{xx} & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{yx} & I_{yy} & -I_{yz} \\ -my_g & mx_g & 0 & -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \quad (3.23)$$

Devido à rotação do veículo subaquático, durante a missão, aparece um efeito de Coriolis, que pode ser representado segundo a equação 3.24 [47].

$$C_{RB}(v) = \begin{bmatrix} mS(v_2) & -mS(v_2)S(r_g^b) \\ mS(r_g^b)S(v_2) & mS(I_b v_2) \end{bmatrix} \quad (3.24)$$

onde r_g^b é a matriz da posição do CG. O vetor v_2 é o vetor velocidade e I_b é o matriz do momento de inercia e S representa a matriz antissimétrica. Fossen [47] mostrou que multiplicando a parte superior da matriz apresentada na equação 3.24 pelo vetor de velocidade, obtém-se a seguinte expressão para o efeito de Coriolis:

$$C_{RB}(v)v = \begin{bmatrix} m(qw - rv) \\ m(ru - pw) \\ m(pv - qu) \\ qr(I_{zz} - I_{yy}) \\ rp(I_{xx} - I_{zz}) \\ qp(I_{yy} - I_{xx}) \end{bmatrix} \quad (3.25)$$

Passando para as forças hidrodinâmicas (τ_{hidro}) que inclui o *drag* hidrodinâmico ($D(v)v$), os efeitos Coriolis-centripetal ($C_A(v)v$) e os efeitos de inercia ($M_A(\dot{v})$) tal

como mostrado na equação 3.26 [47].

$$\tau_{hidro} = -D(v)v - C_A(v)v - M_A(\dot{v}) \quad (3.26)$$

Assumindo que o veículo não funcionará a velocidades elevadas, é possível simplificar os componentes de τ_{hidro} [47]

$$M_A = - \begin{bmatrix} X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{\dot{r}} \end{bmatrix} \quad (3.27)$$

$$C_{Av} = - \begin{bmatrix} Y_{\dot{v}}vr - Z_{\dot{w}}wq \\ Z_{\dot{w}}wp - X_{\dot{u}}ur \\ X_{\dot{u}}uq - Y_{\dot{v}}vp \\ (Y_{\dot{v}} - Z_{\dot{w}})vw + (M_{\dot{q}} - N_{\dot{r}})qr \\ (Z_{\dot{w}} - X_{\dot{u}})uw + (N_{\dot{r}} - K_{\dot{p}})pr \\ (X_{\dot{u}} - Y_{\dot{v}})uv + (K_{\dot{p}} - M_{\dot{q}})pq \end{bmatrix} \quad (3.28)$$

A força de *drag* é composta por duas componentes: uma linear e outra quadrática. D_{linear} é uma matriz diagonal com os componentes de *drag* lineares e $D_{quadratica}$ é uma matriz diagonal com os componentes de *drag* quadráticos. Assim a componente $D(v)v$ fica o seguinte:

$$D(v)v = - \begin{bmatrix} (X_u + X_{u|u}|u|)u \\ (Y_v + Y_{v|v}|v|)v \\ (Z_w + Z_{w|w}|w|)w \\ (K_p + K_{p|p}|p|)p \\ (M_q + M_{q|q}|q|)q \\ (N_r + N_{r|r}|r|)r \end{bmatrix} \quad (3.29)$$

De seguida, as forças hidrostática. Estas dividem-se em duas: a força de *buoyancy* e a força gravitacional ou peso.

$$g(\eta) = \begin{bmatrix} (W - B)\sin(\theta) \\ -(W - B)\cos(\theta)\sin(\phi) \\ -(W - B)\cos(\theta)\cos(\phi) \\ -(y_gW - y_bB)\cos(\theta)\cos(\phi) + (z_gW - z_bB)\cos(\theta)\sin(\phi) \\ (z_gW - z_bB)\sin(\theta) + (x_gW - x_bB)\cos(\theta)\cos(\phi) \\ -(x_gW - x_bB)\cos(\theta)\sin(\phi) + (y_gW - y_bB)\sin(\theta) \end{bmatrix} \quad (3.30)$$

onde $W = mg$ é o peso do veículo, $B = \rho g V$ é a força *buoyancy*. A posição do centro de gravidade é definido por $r_g^b = [x_g \ y_g \ z_g]^T$ e $r_b^b = [x_b \ y_b \ z_b]^T$ é a posição de centro de *buoyancy*.

Por fim, a última componente do modelo dinâmico envolve o cálculo das forças dos atuadores. Para a obtenção desta componente é necessário obter outros três parâmetros: a matriz de coeficientes dos motores, a matriz de alocação dos motores e os coeficientes de perda de força. Na matriz de coeficientes dos motores estabelece a dependência entre o RPM do motor e a força que o motor realiza. A matriz de alocação dos motores especifica a configuração dos motores no veículo. Por fim, os coeficientes de perda de força é o coeficiente de perda para cada motor [48].

No final da obtenção de cada parâmetro, é obtido uma equação idêntica à 3.31; obtendo a contribuição de cada motor para a força e o torque aplicado ao AUV.

$$T_i = \begin{bmatrix} T_x \\ T_y \\ T_z \\ T_\phi \\ T_\theta \\ T_\psi \end{bmatrix} \quad (3.31)$$

Capítulo 4

Arquitetura do sistema

Este capítulo encontra-se dividido entre dois grandes pontos: *Hardware* e *Software*. Na secção de *Hardware* é exposto os sensores utilizados em cada um dos integrantes de toda a manobra de docagem. Na secção de *Software* é apresentado a solução desenvolvida para eliminar o problema proposto: a localização relativa e absoluta. Todos os sensores a utilizar no sistema de docagem são de baixo custo.

4.1 Sistemas Robóticos integrantes

Nesta primeira secção achou-se por bem explicar os restantes sistemas integrantes na realização da manobra, para além da estação de docagem.

4.1.1 Bóia

A boiá, que se encontra encontra a flutuar no mar, caracteriza-se por ser uma estrutura cilíndrica com 13 metros de altura.

Esta encontra-se presa por um cabo ao sistema de docagem. Este cabo para além de ter a função de carregar o Turtle, também permite a transferência de informação entre a bóia e a doca.

Os sensores que integram a bóia são os seguintes:

- *Transceptor* USBL;
- Sistema *Global Navigation Satellite System* (GNSS);

4.1.2 Lander Robótico Turtle

Este sistema autónomo será a sua terceira versão, não sendo ele o principal foco deste projeto. Por isso, todo o seu sistema de localização e controlo já deverá se encontrar desenvolvido, não dependendo deste trabalho.

Ao que esta dissertação interesse, é o *pinger* USBL presente do *lander*. Será também necessário colocar um anel de *Light-Emitting Diodes* (LED's) no Turtle, de forma a que seja possível determinar a localização Turtle em relação ao sistema de docagem.

4.2 Localização da doca no referencial de navegação

Para a solução da localização absoluta da doca foi desenhado e pensado o diagrama apresentado na figura 4.1. Neste é exposto o *hardware* e os blocos de *software* a utilizar para o desenvolvimento da solução.

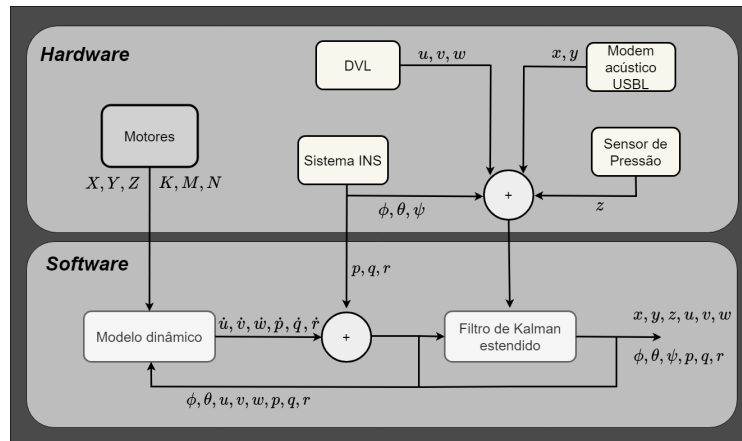


Figura 4.1: Arquitetura geral da solução para localização absoluta do sistema aquático

Começando pelo *hardware*, neste tipo de aplicações a obtenção da atitude e da posição do veículo obtém-se, tipicamente, a partir de sensores inerciais, através da dupla integração. No entanto, por esse método, a informação sobre a posição rapidamente se degrada. Para solucionar esta adversidade recorre-se a outro tipo de sensores.

Um possível complemento poderia passar pela adição de um DVL, bastante referenciado na literatura. Porém este não obtém a posição diretamente, mas sim através da integração da velocidade, tornando o sistema sensível a erros. Por exemplo, caso haja um erro no tempo de aquisição da medida, na rotação do sensor para o referencial do veículo ou no alinhamento, este seria propagado para a posição.

Deste modo é importante ter um sistema que observe diretamente a posição, de forma a manter os erros em posição limitado. Assim a escolha recaiu sobre um

sistema USBL, onde será possível obter a posição do sistema de docagem em relação à bóia. Sendo a configuração a seguinte: um *transceiver* USBL na bóia e um modem standard na doca. Assim o USBL presente na bóia consegue medir tanto a distância como dois ângulos da posição relativa da doca, no referencial do USBL na bóia. Contudo, este tipo de sensores acústico apresentam uma precisão pobre no plano vertical, aparecendo assim a utilização de um sensor de pressão; Com este é obtido a profundidade do AUV.

Com esta arquitetura de *hardware* será possível observar diretamente 9 dos doze estados a estimar, ficando apenas de fora as velocidades angulares do sistema; apesar que estas podem ser observadas diretamente pelo sistema INS.

Com esta premissa em mente e para além do já abordado sistema USBL e do sensor de pressão, o AUV será capacitado de um DVL, para obter as velocidades lineares, e um sistema INS observando diretamente a sua orientação.

O bloco denominado de motores, é indicativo das forças que os motores exercem na doca, importantes para o modelo dinâmico.

4.2.1 Software

Relativamente ao software, este encontra-se mais em foco na figura 4.2. Aqui é apresentado com maior detalhe para onde cada informação de cada sensor vai no filtro de Kalman.

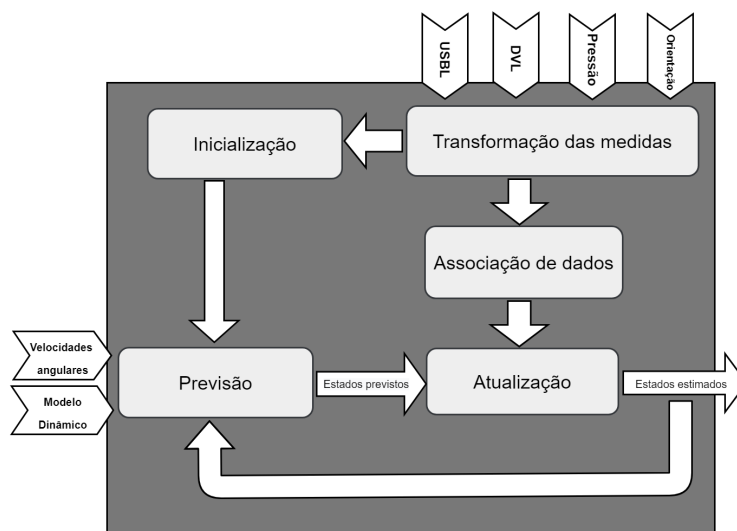


Figura 4.2: Diagrama do bloco filtro de Kalman estendido da arquitetura do *software*

Começando para a parte da previsão, este tem como sinais de controlo as velocidades angulares, obtidas pelo sistema INS, e as acelerações do sistema, calculadas pelo modelo dinâmico.

Os dados do sistema USBL, o DVL, do sensor de pressão e do sistema INS, neste caso a orientação, são transformados para o referencial de navegação, passam ainda por uma camada de associação de dados e por fim, entram na fase de atualização dos estados previstos. A inicialização faz uso destes mesmo dados. Importa referir que a informação USBL, apresentada na figura 4.1, já se encontra compensada em posição e orientação.

Através deste filtro é estimado a posição, a velocidade linear, a orientação e a velocidade angular do sistema de docagem num referencial de navegação NED.

4.2.2 Hardware

Ao nível da sua sensorização esta é composta pelos seguintes sensores:

- Beacon/Pinger USBL;
- Sistema de imagem - câmara;
- Um DVL;
- Sensor de pressão;
- Sistema INS.

O *pinger USBL* utilizado no sistema de docagem é o S2C T 18/34 da EvoLogics. Com a utilização de um *beacon*, quem terá a informação a recolher do sistema USBL será quem possui o *transponder*, neste caso a bóia. Isto poderia ser um problema caso a bóia e o sistema de docagem não se encontrassem ligados através de um cabo. Na eventualidade disso não ser verdade, seria necessário transmitir a informação por um modem acústico.

De entre as várias especificações deste *pinger* destaca-se o seu alcance máximo de 3500 m, uma frequência de 18 - 34 kHz e um *bit error rate* de menos de 10^{-10} . A máxima profundidade que opera depende do seu revestimento, caso seja aço inoxidável é capaz de funcionar até 2000 m, caso seja *delrin* apenas funciona a uma máxima profundidade de 200 m [7].



Figura 4.3: *Pinger* USBL S2C T 18/34 [7]

O DVL utilizado na doca é o A50 da WaterLinked. Este DVL destaca-se pelo seu pequeno tamanho, importante pois o sistema de docagem não possui muito espaço para a alocação dos sensores.

Para além das suas características físicas, este pequeno dispositivo é capaz de medir até distâncias até 50 m, sendo que a sua performance para mais de 35 m depende de vários fatores. Possui 4 feixes montados, cada um com um ângulo de 22.5° com o eixo central.

Ademais dispõe de uma resolução na velocidade de 0.1 mm/s, e uma precisão de $\pm 1.01\%$ na versão standard e $\pm 0.1\%$ na versão performance.



Figura 4.4: DVL A50 da WaterLinked [8]

O Cube Orange funcionará como o sistema INS no sistema de docagem. O processador ARM de 32bit STM32H743 Cortex-M7 é o principal componente deste sistema [9]. O sistema de localização a desenvolver nesta dissertação fará parte do software a integrar esta placa. Assim como o sistema de controlo e algoritmo de navegação do sistema de docagem.

No que se refere a sensores, na sua origem, o Cube Orange possui três IMUs, criando um sistema redundante, sendo eles os seguintes [9]:

- ICM 20649 com acelerómetro e giroscópio, barómetro MS5611 na placa de base;
- Numa placa isolada de vibrações o IMU ICM20602 , o IMU/MAG ICM20948 e o barómetro MS5611;

Com os sensores que o Cube abarca, este permite obter a orientação do sistema de docagem.



Figura 4.5: *Cube Orange* sistema para obtenção dos dados inerciais [9]

4.3 Localização Turtle relativa à doca

Abordando a localização relativa, a localização do turtle em relação à doca, neste é também utilizado um filtro EKF. Onde apenas é utilizado o sistema USBL, Turtle $\leftrightarrow$ Bóia, e o feedback de uma câmara, montada na parte inferior da doca.

Assim a arquitetura geral para esta solução, figura 4.6, abarca, no *hardware*, apenas o sistema USBL e a câmara, enquanto na parte do software, o EKF volta a ser o escolhido. A diferença para o apresentado anteriormente é que nesta solução é utilizado um modelo de velocidade constante na parte da previsão. Na segunda parte do filtro, as observações podem ser do sistema USBL ou o feedback da câmara. Até ao momento que a câmara consegue observar o sistema de luzes do Turtle, as observações do filtro são as obtidas pelo sistema USBL, quando a câmara deteta o sistema de luzes, passa a conseguir determinar a posição do Turtle em relação à câmara. A partir desse momento, o filtro utiliza os dados obtidos do bloco da câmara na atualização dos estados previstos.

A possibilidade de dividir a missão de docagem em duas, momento em que o Turtle não se encontra ao alcance da câmara e momento em que se encontra, permite simplificar a solução desenhada. No primeiro momento, o sistema USBL é primordial, e apesar desta sistema não disponibilizar informação de forma contínua, o importante é ter uma estimação da zona onde se encontra o Turtle. Aqui a precisão não é importante. No momento da docagem em si, a precisão passa a ser importante, e por isso se dá primazia ao feedback da câmara.

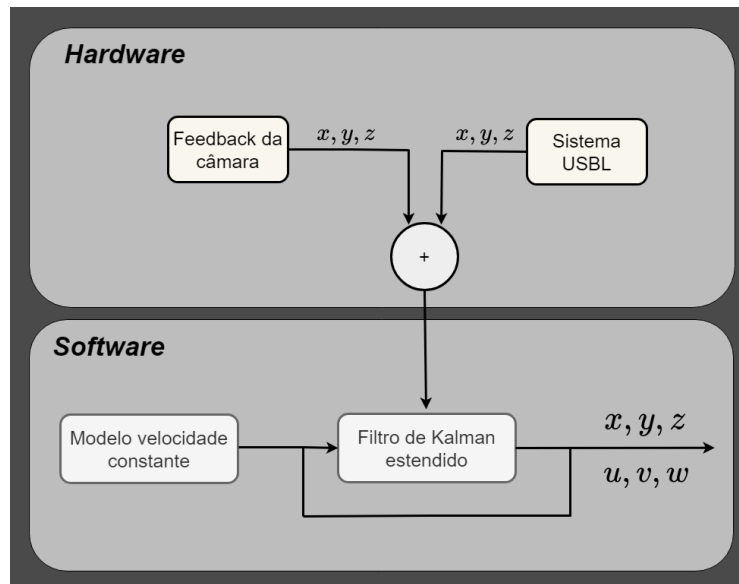


Figura 4.6: Arquitetura de geral da solução para localização absoluta do sistema aquático

4.3.1 Hardware

O sistema USBL utilizado no lander robótico é o S2C R 18/34D da EvoLogics [49]. Este, apesar de ser um modem acústico, pode funcionar de forma idêntica a um *beacon*. Respondendo assim aos pedidos do USBL colocado na bóia flutuante. Assim a distância e os dois ângulos obtidos serão referentes à bóia. Por essa razão é necessário um bloco de software que trate de transformar esses dados para serem relativos ao sistema de docagem.

Apesar de ainda não estar decidida qual câmara a ser utilizada na estação de docagem, esta deverá ser uma idêntica aos módulos de câmara para a coral [50] ou para a raspberry [51]. No caso do módulo para a coral, esta câmara possui 5 Megapixels com conexão através de uma interface MIPI-CSI. Este módulo contém um sensor Omnivision OV5645 SoC, com as seguintes características:

- Comprimento focal: 2.5 mm;
- Campo de visão: $84.0^\circ / 87.6^\circ$;
- Focagem automática;
- Balanceamento de brancos automático;
- calibração dos níveis de pretos automático.

4.3.2 Software

No que diz respeito ao software, quando comparando com a solução anterior, esta é mais simples, figura 4.7.

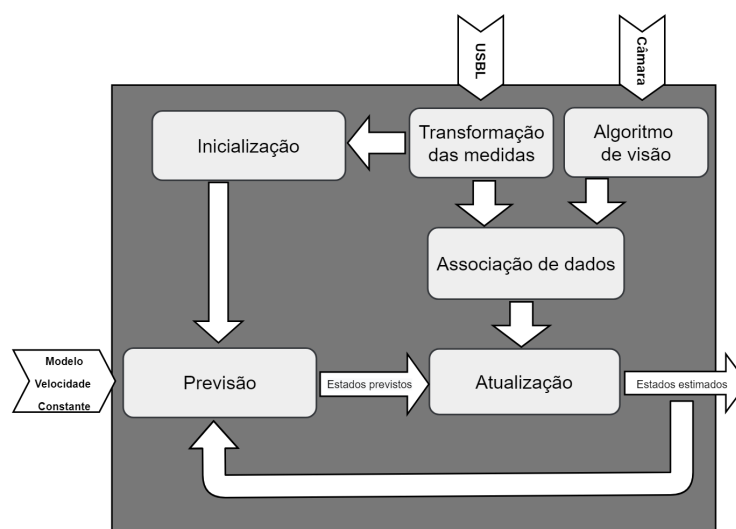


Figura 4.7: Diagrama do bloco filtro de Kalman estendido da figura 4.1

A inicialização do filtro deverá ser realizada pela primeira medida obtida pelo sistema USBL. O modelo de velocidade constante é utilizado na fase de previsão do EKF. Para a fase de atualização, o desenvolvimento de um algoritmo que gere qual das medidas deve ser utilizada é imprescindível. A ideia passa pela utilização da informação do USBL até ao momento que a câmara consiga detetar o anel de LED's presente no Turtle.

O bloco de associação de dados torna-se importante pois é neste bloco onde se decide qual dos dados a utilizar. A utilização dos dados obtidos pelo USBL é primordial até ao momento que a câmara comece a observar o Turtle. Quando é possível obter dados da câmara, os dados do USBL são preteridos em favor do feedback da câmara. O bloco de associação de dados é responsável por esta dinâmica.

Esta solução permitirá ter um maior controlo na última fase da manobra de docagem, uma vez que a câmara observará diretamente a relação entre a doca e o Turtle. Solucionando assim um dos objetivos referidos em 1.2.

Capítulo 5

Implementação

Nesta secção é exposto toda a implementação realizada para esta dissertação de forma a responder aos objetivos propostos.

5.1 Filtro de Kalman estendido - localização doca

Tal como apresentado em 4.2, a escolha para a fusão sensorial recaiu sobre a implementação de um filtro de Kalman estendido. A decisão deve-se essencialmente devido ao modelo de processo não ser linear.

Posto isto, o vetor de estados a estimar é constituído por:

$$\hat{x}_k = \begin{bmatrix} x & y & z & u & v & w & \phi & \theta & \psi & p & q & r \end{bmatrix}^T \quad (5.1)$$

onde x, y, z corresponde à posição no referencial de navegação, u, v, w é referente à velocidade inercial na referencial de navegação, ϕ, θ, ψ relativo à orientação do AUV e p, q, r corresponde à velocidade angular do sistema. O referencial de navegação escolhido foi o referencial NED numa posição local de referência para a bóia.

Para a implementação deste filtro primeiramente é necessário realizar a sua inicialização. Estes estados iniciais $\hat{x}_0$ são obtidos através da informação inicial dos sensores.

Ou seja, a posição em x e y é obtido com base na primeira informação transmitida pelo sistema USBL, após a aquisição dessa informação inicializa-se os restantes estados observáveis. Para o cálculo da profundidade é utilizado a equação 5.2:

$$prof = \frac{P - P_0}{\rho g} \quad (5.2)$$

onde *prof* se refere à posição em z , P a pressão medida pelo sensor, P_0 a pressão atmosférica, ρ a densidade de água e g a força gravítica.

As velocidade lineares são obtidas com base no DVL e a orientação ϕ , θ e ψ com a informação recolhida pelo sistema INS.

Assim, os estados iniciais e as suas incertezas são representados na seguinte forma:

$$\hat{x}_0 = \begin{bmatrix} x_{usbl} \\ y_{usbl} \\ z_p \\ u_{dvl} \\ v_{dvl} \\ w_{dvl} \\ \phi_{ins} \\ \theta_{ins} \\ \psi_{ins} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \sigma_{usbl} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{usbl} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{dvl} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{dvl} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{dvl} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{ins} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{ins} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{ins_\psi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_q & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_r \end{bmatrix} \quad (5.3)$$

A opção por esta solução deve-se ao facto da taxa de dados do DVL, do INS e do sensor de pressão ser constante e por isso ser fácil de aceder à mesma, enquanto o mesmo não acontece para a informação proveniente do sistema USBL. Por isso, optou-se por começar a fusão sensorial após a receção da primeira informação do USBL. Importa referir que a incerteza deve estar coerente com o possível erro inicial.

5.1.1 Fase de previsão

Após a inicialização, o filtro avança para a fase de previsão. Nesta fase optou-se por desenvolver um modelo dinâmico idêntico à realidade, com base no modelo apresentado em 3.5. Para além das acelerações calculadas, com recurso ao modelo, adiciona-se ao vetor de controlo os dados da velocidade angular. Estes são obtidos diretamente a partir do sistema INS. Com isto o vetor de controlo é apresentado da seguinte maneira:

$$u = \begin{bmatrix} dyn_{\dot{u}} & dyn_{\dot{v}} & dyn_{\dot{w}} & dyn_{\dot{p}} & dyn_{\dot{q}} & dyn_{\dot{r}} & gyro_p & gyro_q & gyro_r \end{bmatrix}^T \quad (5.4)$$

onde $dyn_{\dot{u}}$, $dyn_{\dot{v}}$ e $dyn_{\dot{w}}$ representam as acelerações lineares do AUV, $dyn_{\dot{p}}$, $dyn_{\dot{q}}$ e $dyn_{\dot{r}}$ as acelerações angulares e $gyro_p$, $gyro_q$ e $gyro_r$ as velocidades angulares medidas pelo sistema INS.

De seguida, o modelo de processo é definido por:

$$\dot{x} = f(x, u) \quad (5.5)$$

$$\begin{bmatrix} x_k \\ y_k \\ z_k \\ u_k \\ v_k \\ w_k \\ \phi_k \\ \theta_k \\ \psi_k \\ p_k \\ q_k \\ r_k \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} x_{k-1} + u_{k-1}dt \\ y_{k-1} + v_{k-1}dt \\ z_{k-1} + w_{k-1}dt \end{bmatrix} + R \begin{bmatrix} \frac{(dyn_{\dot{u}} + \sigma_{dyn})dt^2}{2} \\ \frac{(dyn_{\dot{v}} + \sigma_{dyn})dt^2}{2} \\ \frac{(dyn_{\dot{w}} + \sigma_{dyn})dt^2}{2} \end{bmatrix} \\ \begin{bmatrix} u_{k-1} \\ v_{k-1} \\ w_{k-1} \end{bmatrix} + R \begin{bmatrix} (dyn_{\dot{u}} + \sigma_{dyn})dt \\ (dyn_{\dot{v}} + \sigma_{dyn})dt \\ (dyn_{\dot{w}} + \sigma_{dyn})dt \end{bmatrix} \\ \begin{bmatrix} \phi_{k-1} \\ \theta_{k-1} \\ \psi_{k-1} \end{bmatrix} + R \begin{bmatrix} (gyro_p + \sigma_{gyro})dt \\ (gyro_q + \sigma_{gyro})dt \\ (gyro_r + \sigma_{gyro})dt \end{bmatrix} + R \begin{bmatrix} \frac{(dyn_{\dot{p}} + \sigma_{dyn})dt^2}{2} \\ \frac{(dyn_{\dot{q}} + \sigma_{dyn})dt^2}{2} \\ \frac{(dyn_{\dot{r}} + \sigma_{dyn})dt^2}{2} \end{bmatrix} \\ R \begin{bmatrix} (gyro_p + \sigma_{gyro}) \\ (gyro_q + \sigma_{gyro}) \\ (gyro_r + \sigma_{gyro}) \end{bmatrix} + R \begin{bmatrix} (dyn_{\dot{p}} + \sigma_{dyn})dt \\ (dyn_{\dot{q}} + \sigma_{dyn})dt \\ (dyn_{\dot{r}} + \sigma_{dyn})dt \end{bmatrix} \end{bmatrix} \quad (5.6)$$

onde R é a matriz de rotação do referencial do corpo para o referencial da navegação, definida em 3.5, dt o intervalo de tempo entre o momento anterior $k - 1$ e o momento atual k de filtragem, o $\sigma_{dy_{vx}}$, o $\sigma_{dy_{vy}}$ e o $\sigma_{dy_{vz}}$ o ruído, relativo às velocidades lineares, e o $\sigma_{dy_{ax}}$, o $\sigma_{dy_{ay}}$ e o $\sigma_{dy_{az}}$ o ruído, relativo às velocidades angulares, associado ao modelo dinâmico, por fim, o o σ_{gyro} é o ruído associado ao sistema INS.

Relativamente à covariância do modelo de processo, esta é representada pela matriz Q_k dada por:

$$Q_k = \begin{bmatrix} \sigma_{dy_{vx}}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{dy_{vy}}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{dy_{vz}}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{dy_{ax}}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{dy_{ay}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{dy_{az}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{gyro}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{gyro}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{gyro}^2 \end{bmatrix} \quad (5.7)$$

O facto do modelo não ser linear implica a utilização do filtro EKF exposto.

Devido a esta não linearidade é necessário linearizar o modelo com recurso ao jacobiano. Este passo implica a linearização da matriz de transição de estados e da matriz que relaciona os estados com o ruído. Posto isto, a matriz F_k que representa a matriz de transição de estados linearizada é dada por:

$$F_k = \frac{\partial f}{\partial x}(\dot{x}_{k|k-1}, u_k, 0) \quad (5.8)$$

$$F_k = \begin{bmatrix} 1 & 0 & 0 & dt(-q_3^2 - q_2^2 + q_1^2 + q_0^2) & 2dt(q_1q_2 - q_0q_3) & 2dt(q_1q_3 + q_0q_2) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2dt(q_0q_3 - q_1q_2) & dt(-q_3^2 + q_2^2 - q_1^2 + q_0^2) & 2dt(q_2q_3 - q_0q_1) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2dt(q_1q_3 - q_0q_2) & 2dt(q_2q_3 + q_0q_1) & dt(q_3^2 - q_2^2 - q_1^2 + q_0^2) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt(-q_3^2 - q_2^2 + q_1^2 + q_0^2) & 2dt(q_1q_2 - q_0q_3) & 2dt(q_1q_3 + q_0q_2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2dt(q_0q_3 - q_1q_2) & dt(-q_3^2 + q_2^2 - q_1^2 + q_0^2) & 2dt(q_2q_3 - q_0q_1) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2dt(q_1q_3 - q_0q_2) & 2dt(q_2q_3 + q_0q_1) & dt(q_3^2 - q_2^2 - q_1^2 + q_0^2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

Por outro lado, a matriz G_k que correlaciona os estados com ruído linearizada é ditada por:

$$G_k = \frac{\partial f}{\partial \sigma}(\dot{x}_{k|k-1}, u_k, 0) \quad (5.10)$$

$$G_k = \begin{bmatrix} \frac{1}{2}dt^2(-q_3^2 - q_2^2 + q_1^2 + q_0^2) & dt^2(q_1q_2 - q_0q_3) & \frac{1}{2}dt^2(2(q_1q_3 + q_0q_2)) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ dt^2(q_0q_3 - q_1q_2) & \frac{1}{2}dt^2(-q_3^2 + q_2^2 - q_1^2 + q_0^2) & dt^2(q_2q_3 - q_0q_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ dt^2(q_1q_3 - q_0q_2) & dt^2(q_2q_3 + q_0q_1) & \frac{1}{2}dt^2(q_3^2 - q_2^2 - q_1^2 + q_0^2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ dt(-q_3^2 - q_2^2 + q_1^2 + q_0^2) & 2dt(q_1q_2 - q_0q_3) & 2dt(q_1q_3 + q_0q_2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2dt(q_0q_3 - q_1q_2) & dt(-q_3^2 + q_2^2 - q_1^2 + q_0^2) & 2dt(q_2q_3 - q_0q_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2dt(q_1q_3 - q_0q_2) & 2dt(q_2q_3 + q_0q_1) & dt(q_3^2 - q_2^2 - q_1^2 + q_0^2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}dt^2(-q_3^2 - q_2^2 + q_1^2 + q_0^2) & dt^2(q_1q_2 - q_0q_3) & \frac{1}{2}dt^2(2(q_1q_3 + q_0q_2)) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & dt^2(q_0q_3 - q_1q_2) & \frac{1}{2}dt^2(-q_3^2 + q_2^2 - q_1^2 + q_0^2) & dt^2(q_2q_3 - q_0q_1) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & dt^2(q_1q_3 - q_0q_2) & dt^2(q_2q_3 + q_0q_1) & \frac{1}{2}dt^2(q_3^2 - q_2^2 - q_1^2 + q_0^2) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & dt & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & dt & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & dt & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.11)$$

Deste modo, e como as várias matrizes apresentadas anteriormente, a matriz de covariância do modelo de processo é calculada através da seguinte equação:

$$P_{k|k-1} = F_k P - k - 1 F_k^T + G_k Q_k G_k^T \quad (5.12)$$

5.1.2 Fase de atualização

A fase de atualização tem como principal objetivo a correção dos estados previstos, obtidos na fase da previsão, com base no modelo das observações. Este modelo é linear, uma vez que os sensores observam diretamente os estados. Assim, os sensores utilizados são os seguintes:

- USBL para observação da posição x e y ;
- Sensor de pressão para a posição em z ;
- DVL para observação das velocidades lineares, u , v e w ;

- Sistema INS para a orientação, ϕ , θ e ψ .

Posto isto, o modelo de observação é definido pela seguinte expressão:

$$z_k = Hx_k + \eta_k \quad (5.13)$$

onde z_k representa a observação no instante k , η_k o ruído associado a essa observação e a matriz H é a matriz do modelo de observações. A forma desta matriz depende diretamente de quais os sensores que possuem uma medição no instante k , isto devido às distintas taxa de dados dos sensores. A título de exemplo, o sistema USBL apresenta uma taxa de dados bastante mais pequena do que o DVL. Isso terá que ser contemplado no modelo de observações e por isso, primeiramente é definido a matriz H associada a cada um dos sensores.

Importa realçar que independentemente das medidas dos sensores serem referentes a outro referencial, nesta fase todas as observações utilizadas já se encontram relativas ao referencial de navegação.

Começando pelo sistema USBL, tal como referido, aqui é observado a posição em x e y e por isso, a matriz H_{USBL} é definida por:

$$H_{USBL} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.14)$$

A covariância associada é dada por:

$$R_{USBL} = \begin{bmatrix} \sigma_{USBLx}^2 & 0 \\ 0 & \sigma_{USBLy}^2 \end{bmatrix} \quad (5.15)$$

A matriz do modelo de observações associada ao sensor pressão é dada por:

$$H_{pressao} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.16)$$

Já a sua covariância associada é dada por:

$$R_{pressao} = \sigma_P^2 \quad (5.17)$$

Relativamente ao DVL, este permite ler as velocidade lineares em x , y e z , e assumindo que estas já se encontram relativas ao referencial de navegação. A matriz H_{DVL} é dada por:

$$H_{DVL} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.18)$$

Sendo que a sua covariância associada é fornecida por:

$$R_{DVL} = \begin{bmatrix} \sigma_{DVL}^2 & 0 & 0 \\ 0 & \sigma_{DVL}^2 & 0 \\ 0 & 0 & \sigma_{DVL}^2 \end{bmatrix} \quad (5.19)$$

Por fim, a matriz relativa à orientação fornecida pelo sistema INS é dada por:

$$H_{ori} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (5.20)$$

Com a sua matriz de covariância associada dada por:

$$R_{ori} = \begin{bmatrix} \sigma_{ori}^2 & 0 & 0 \\ 0 & \sigma_{ori}^2 & 0 \\ 0 & 0 & \sigma_{ori}^2 \end{bmatrix} \quad (5.21)$$

Tal como referido anteriormente, os sensores utilizados possuem diferentes taxas de dados, ou seja, no momento que se realizará o *update*, dos estados e da covariância, diferentes dados poderão estar disponível. Por esta razão a matriz H e R são variáveis, sendo que será contemplado a necessidade de retirar a matriz H e R do sensor não utilizado naquele momento, por exemplo. Caso todos os sensores sejam utilizados a matriz H e R ficam com a seguinte forma:

$$H_k = \begin{bmatrix} H_{USBL} \\ H_{pressao} \\ H_{DVL} \\ H_{ori} \end{bmatrix} \quad R_k = \begin{bmatrix} R_{USBL} & 0 & 0 & 0 \\ 0 & R_{pressao} & 0 & 0 \\ 0 & 0 & R_{DVL} & 0 \\ 0 & 0 & 0 & R_{ori} \end{bmatrix} \quad (5.22)$$

Com base no descrito acima, a covariância estimada é calculada pela seguinte expressão:

$$P_k = P_{k|k-1} - K_k S_k K_k^T \quad (5.23)$$

onde K é o ganho de Kalman e S é a inovação associada à covariância, estas duas variáveis são calculadas da seguinte forma:

$$S_k = H_k P_{k|k-1} H_k^T + R_k \quad K_k = P_{k|k-1} H_k S_k^{-1} \quad (5.24)$$

A correção aos estados previstos, que são os estados de output da fase de previsão, é realizada através da seguinte expressão:

$$x_k = x_{k|k-1} + K_k (z_k - H_k x_{k|k-1}) \quad (5.25)$$

obtendo assim os estados estimados.

Um dos dados importantes para validar o funcionamento do filtro é a inovação associada às observações, esta permite perceber a relação entre as observações e os estados previstos, e que é expressa por:

$$\tilde{y}_k = z_k - H_k x_{k|k-1} \quad (5.26)$$

5.2 Filtro de Kalman estendido - localização doca ↔ Turtle

Nesta secção é apresentado o filtro que será capaz de localizar o Turtle no referencial da doca. Fundamental para que a manobra de docagem se realize com sucesso.

Posto isto, o vetor de estados a estimar é constituído por:

$$\hat{x}_j = \begin{bmatrix} x & y & z & u & v & w \end{bmatrix}^T \quad (5.27)$$

onde x, y, z corresponde à posição da doca relativa ao turtle e u, v, w corresponde às velocidades lineares entre a doca e o turtle.

Tal como na fusão sensorial anterior, neste caso, é importante começar pela inicialização do filtro. Aqui a inicialização é mais simples, até pelo facto da existência de menos estados a estimar. A solução é idêntica à apresentada em 5.1, onde a primeira informação obtida através do sistema USBL permite o início da filtragem. Aqui os dados do sistema USBL são utilizados para obter a posição do Turtle em x, y e z em relação à doca.

Com isto, os estados iniciais e as suas incertezas são representados da seguinte forma:

$$\hat{x}_0 = \begin{bmatrix} x_{usbl} \\ y_{usbl} \\ z_{usbl} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \sigma_{usbl} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{usbl} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{usbl} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_u & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_v & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_w \end{bmatrix} \quad (5.28)$$

Tal como referido na arquitetura da solução apresentada, os sensores a utilizar nesta solução são o sistema USBL e a câmara. Como inicialmente o Turtle não se encontra no alcance do sistema de visão optou-se por utilizar os dados do USBL para inicializar-se o filtro. As velocidades lineares são inicializadas a 0.

5.2.1 Fase de previsão

Com a inicialização, segue-se a fase da previsão. Diferente do filtro anterior, neste é utilizado um modelo de velocidade constante. A escolha neste modelo deve-se ao

facto da sua simplicidade e porque a aplicação deste filtro EKF assim o permite. O caso de apenas ser necessário estimar com precisão a posição x, y, z é também um dos fatores de se escolher o modelo de velocidade constante. Assim, o modelo de processo é definido por:

$$\dot{x} = f(x) \quad (5.29)$$

$$\begin{bmatrix} x_j \\ y_j \\ z_j \\ u_j \\ v_j \\ w_j \end{bmatrix} = \begin{bmatrix} x_{j-1} + u_{j-1}dt \\ y_{j-1} + v_{j-1}dt \\ z_{j-1} + w_{j-1}dt \\ u_{j-1} + dt\sigma_{accel} \\ v_{j-1} + dt\sigma_{accel} \\ w_{j-1} + dt\sigma_{accel} \end{bmatrix} \quad (5.30)$$

Quando não existe informação acerca dos controlos ou dinâmica do sistema, tipicamente considera-se um modelo de velocidade constante. Apesar disso, é utilizado uma perturbação na velocidade, sendo este o parâmetro que permite tornar o filtro mais responsivo a dinâmicas mais elevadas ou suavizar mais a trajetória. Este parâmetro é representado por σ_{accel} .

Relativamente à covariância do modelo de processo, esta é representada pela matriz Q_k dada por:

$$Q_k = \begin{bmatrix} \sigma_{accel}^2 & 0 & 0 \\ 0 & \sigma_{accel}^2 & 0 \\ 0 & 0 & \sigma_{accel}^2 \end{bmatrix} \quad (5.31)$$

Tal como apresentado anteriormente, é necessário obter a matriz de transição de estados linearizada. Esta é obtida através da equação 5.8 e dada para matriz 5.32.

$$F_k = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & dt \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.32)$$

A matriz G_k que correlaciona os estados com ruído linearizada é obtida pela equação 5.10 e apresentada da seguinte forma:

$$G_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{bmatrix} \quad (5.33)$$

Por fim, e de igual forma ao filtro apresentado anteriormente, a matriz de covariância do modelo de processo é calculado com a equação 5.12.

5.2.2 Fase de atualização

A fase de atualização desta solução apenas utiliza como observações a informação proveniente do USBL ou o feedback da câmara. A dinâmica de funcionamento desta fase difere da apresentada em 5.1.2. Nesta existem dois sensores diferentes capazes de observar os mesmos estados, posição em x , y e z , sendo por isso necessário dividir qual dos sensores é utilizado para a atualização. Assim, o implementado consiste no uso do sistema USBL até haver informação suficiente na câmara capaz de obter os mesmos dados que o USBL permite. Com esta solução, o sistema de docagem quando afastado do Turtle utilizará o sistema USBL, que é um sistema com menor precisão que a câmara. Isto porque o cálculo da posição relativa Doca $\leftrightarrow$ Turtle é obtido através da posição relativa Doca $\leftrightarrow$ Bóia e da posição relativa Bóia $\leftrightarrow$ Turtle, acumulando assim mais fontes de erros. Na fase em que o sistema de visão já consegue calcular a posição x, y, z , os dados do USBL são preteridos em favor dos dados provenientes da câmara. Esta opção deve-se ao facto de nesta fase ser necessário um fluxo de dados constante e menos ruidosos, o que acontece com os dados provenientes do sistema de visão.

Começando pelo sistema USBL, a matriz H_{USBL} é definida por:

$$H_{USBL} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (5.34)$$

Sendo que a covariância associada é dada por:

$$R_{USBL} = \begin{bmatrix} \sigma_{USBL_x}^2 & 0 & 0 \\ 0 & \sigma_{USBL_y}^2 & 0 \\ 0 & 0 & \sigma_{USBL_z}^2 \end{bmatrix} \quad (5.35)$$

Relativamente à câmara, a matriz do modelo de observações, H_{camara} , é definida por:

$$H_{camara} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (5.36)$$

A covariância associada à informação proveniente da câmara é dada por:

$$R_{camara} = \begin{bmatrix} \sigma_{camara_x}^2 & 0 & 0 \\ 0 & \sigma_{camara_y}^2 & 0 \\ 0 & 0 & \sigma_{camara_z}^2 \end{bmatrix} \quad (5.37)$$

Assim, dependendo da informação a utilizar nesta fase, a matriz H e R ficam idênticas às matrizes apresentadas acima.

Para o cálculo da covariância estimada e para a correção dos estados previstos, é utilizado às equações 5.23 e 5.25, respetivamente.

5.3 Modelo dinâmico

O modelo dinâmico apresentado na secção 3.5 é um bloco importante na solução apresentada.

Tabela 5.1: Parâmetros físicos do sistema de docagem

Parâmetros	Valor
centro de gravidade	$[0 \ 0 \ 0]$
centro de flutuabilidade	$[0 \ 0 \ 0.1]$
Z_b	$[0 \ 0 \ 0.1]$
Comprimento	$0.783686 \ m$
Largura	$0.66 \ m$
Altura	$0.538752 \ m$
Massa	$52.26 \ kg$
Volume	$0.05 \ m^3$
ρ	$1025 \ kg/m^3$
g	$9.81 \ m/s^2$
W	$512.6706 \ N$
B	$502.7625 \ N$
I_{xx}	$1.27 \ kg/m^2$
I_{xy}	$-0.01 \ kg/m^2$
I_{xz}	$0.15 \ kg/m^2$
I_{yx}	$-0.01 \ kg/m^2$
I_{yy}	$1.64 \ kg/m^2$
I_{yz}	$-0.01 \ kg/m^2$
I_{zx}	$0.15 \ kg/m^2$
I_{zy}	$-0.01 \ kg/m^2$
I_{zz}	$2.1 \ kg/m^2$

Tabela 5.2: Coeficientes do sistema de docagem

Parâmetros	Valor
$X_{\dot{u}}$	0
$Y_{\dot{v}}$	0
$Z_{\dot{w}}$	0
$K_{\dot{p}}$	0
$M_{\dot{q}}$	0
$N_{\dot{r}}$	0
X_u	-3.866
Y_v	-4.169
Z_w	-1.051
K_p	-17.520
M_q	-30.868
N_r	-104.285
$X_{u u }$	-59.203
$Y_{v v }$	-65.921
$Z_{w w }$	-184.912
$K_{p p }$	-43.800
$M_{q q }$	-77.171
$N_{r r }$	-52.142

O modelo dinâmico, tal como descrito em 3.5, necessita não só dos parâmetros descritos nas tabelas 5.1 e 5.2 mas também da velocidade e dos ângulos roll e pitch do momento anterior, $k-1$, do sistema de docagem. Com esses dados é possível estimar as acelerações do sistema em k . O calculo é realizado através da equação 5.38.

$$accel = \frac{D(v)v + g(\eta) + C_{RB} + T_k}{M_{RB}} \quad (5.38)$$

De todas as variáveis apresentadas, apenas relativamente à matriz T_k não é apresentada a sua forma.

A matriz de alocação dos motores depende da posição e orientação de cada *thruster*, esta permite transformar a força que cada motor realiza em força aplicada no robô em cada um dos seus graus de liberdade. No sistema de docagem a matriz de alocação dos motores é a seguinte:

$$T_{AM} = \begin{bmatrix} -1.0 & -1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.0 & -1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 1.0 & 1.0 & 1.0 \\ 0 & 0 & -0.03506 & -0.03499 & 0.18596 & -0.18539 & 0.1872 & -0.18575 \\ 0.03291 & 0.03401 & 0 & 0 & -0.18402 & -0.18241 & 0.18717 & 0.18503 \\ -0.27251 & 0.27439 & 0.28166 & -0.27495 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.39)$$

À matriz 5.39 é multiplicado por uma matriz coluna quem contém a informação relativa à força que cada motor se encontra a realizar, obtendo assim a matriz T_k .

5.4 Algoritmo de visão

Este algoritmo tem como principal objetivo a determinação da localização do Turtle relativamente à doca. Como output, deste conjunto de regras, é obtido a posição em x , y e z .

Na tabela 5.3 é apresentado a localização dos LED's no plano da câmara. Esta informação serve para calcular o raio do círculo, formado pelos dezasseis LED's, através do mesmo algoritmo utilizado para o cálculo do raio com base no feedback da câmara.

Tabela 5.3: Localização dos LED's no referencial da câmara

	Em X	Em Y
LED 1	0.1404	0.0601
LED 2	0.1058	0.1120
LED 3	0.0538	0.1467
LED 4	-0.0074	0.1588
LED 5	-0.0686	0.1467
LED 6	-0.1205	0.1120
LED 7	-0.1552	0.0601
LED 8	-0.1674	-0.0012
LED 9	-0.1552	-0.0624
LED 10	-0.1205	-0.1143
LED 11	-0.0686	-0.1490
LED 12	-0.0074	-0.1612
LED 13	0.0538	-0.1490
LED 14	0.1058	-0.1143
LED 15	0.1404	-0.0624
LED 16	0.1526	-0.0012

Os passos a realizar são os seguintes:

- Transformar a imagem de RGB para preto e branco;
- Aplicar um filtro binário;
- Utilização de um algoritmo para deteção de *bulbs*;
- Cálculo das coordenadas do centro de cada um dos *bulbs* na *frame* da câmara;
- Calcular a forma geométrica formada pelos 16 pontos calculados no ponto anterior;
- Cálculo da distância da câmara ao objeto;
- Obtenção da posição X , Y e Z do meio do Turtle em relação à doca.

A aplicação de um filtro binário, com um determinado *threshold*, permite obter uma imagem onde a luz de cada um dos LED's é destacada. A figura 5.1 é apresentado o processo de aplicação dos filtros. Aqui é possível perceber que com o filtro binário a única informação da imagem passa a ser o anel de LED's.

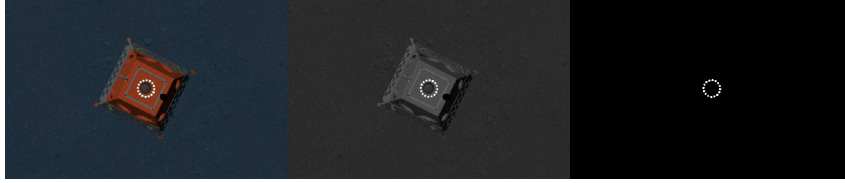


Figura 5.1: Passos realizados na algoritmo de visão: Imagem retirada da câmara, aplicação preto e branco, e aplicação do *threshold*

Com recurso à transformada circular de Hough são obtido os centros (posição x e y) e o raio de cada um dos LED's da imagem binária. De notar que estes centros é dado no referencial da imagem, ou seja, é necessário transformar essa informação para o referencial da câmara. Isso é conseguido pelas seguintes equações:

$$a_x = (centers_x - (x_s/2)) * \frac{FOV_x}{x_s} \quad (5.40)$$

$$a_y = (centers_y - (y_s/2)) * \frac{FOV_y}{y_s} \quad (5.41)$$

onde a_x e a_y são os pontos no referencial da câmara, FOV_x e FOV_y as componentes do parâmetro *field of view* da câmara e por fim, x_s e y_s são a largura e o comprimento da imagem gerada pela câmara.

De seguida estima-se o centro e o raio da figura geométrica formada pelos pontos obtidos anteriormente. Para isto, é utilizado um método de *fit* de uma elipse por mínimos quadrados. A escolha por esta técnica deve-se ao facto de poder haver situações onde figura geométrica detetada seja uma elipse, caso o alinhamento entre os dois sistemas autónomos, não seja perfeito [52].

Por fim, é calculado a posição do centro do Turtle com as seguintes equações:

$$x_{final} = \frac{x_{circulo}}{rad} \quad (5.42)$$

$$y_{final} = \frac{y_{circulo}}{rad} \quad (5.43)$$

$$z_{final} = \frac{radius_{true} * f_{len}}{rad}; \quad (5.44)$$

onde rad é o raio estimado do círculo pelo algoritmo, f_{len} é a distância focal da câmara, em mm, e o $radius_{true}$ é o raio verdadeiro do círculo de LED's, informação conhecida à priori.

Capítulo 6

Resultados

Neste capítulo são apresentados os resultados deste projeto. Relativamente aos resultados são expostos vários gráficos e várias tabelas, que mostram como se comportam os dois filtros durante a manobra de docagem. Para além disso foram realizados alguns testes de forma a simular com melhor precisão o que acontece no mundo real.

6.1 Validação do trabalho desenvolvido

Para a validação dos dois sistemas de localização implementados foi utilizado um simulador capaz de simular a manobra de docagem, descrita em 1.1. O desenvolvimento deste simulador foi realizado por um grupo de alunos composto por mim e mais um colega no âmbito da unidade curricular de Laboratório de Sistemas Multi-robóticos. O foco principal passa pela simulação da manobra de docagem com todos os sensores presentes. Assim os dados para a validação deste projeto são retirados diretamente desse simulador.

O simulador foi desenvolvido com recurso ao *UUV Simulator* [53], que é um pacote *Robot Operating System* (ROS) concebido para o Gazebo. Para além de vários *plugins*, a utilizar diretamente no ambiente de simulação, este pacote contém vários nós ROS para a simulação de diversos meios aquáticos. Ademais é possível controlar e modelar um novo veículo subaquático, como um ROV ou um AUV.

Foi a partir desta última característica do simulador, que foi possível a modelação e o controlo dos sistemas robóticos presentes na manobra, assim como todos os sensores que cada um possui. Os sensores foram simulados de forma a apresentarem

resultados idênticos aos verdadeiros, ou seja, com as mesmas características que os sensores de baixo custo abordados anteriormente.

6.1.1 Simulação dos sensores

Para a simulação dos sensores foi necessário definir alguns parâmetros dos sensores, tais como a sua frequência e ruído. Estes são simulados através de *plugins* previamente desenvolvidos no simulador, podendo alterar os parâmetros para o sensor que queremos simular.

Assim, começando pelo DVL. A este sensor foi definido uma taxa de transmissão de 7 Hz, um ruído em amplitude de 0.03 m/s com um alcance máximo de 50 m. Para além disso, é definido os 4 raios no DVL com um ângulo entre eles de 39 º.

Relativamente ao sistema INS este tem definido uma taxa de aquisição de 50 Hz. Os dados relacionados com a orientação do sistema INS possuem um ruído de 0.1º, os parâmetros relacionados com o acelerómetro e o giroscópio estão expostos na tabela 6.1.

Tabela 6.1: Caracterização de diversos parâmetros do acelerómetro e do giroscópio

	Acelerómetro	Giroscópio
<i>Noise Density</i>	$0.003 \text{ m/s}^2/\sqrt{\text{Hz}}$	$0.003394 \text{ rad}/\sqrt{\text{s}}$
<i>Bias correlation with time</i>	300.0 s	1000.0 s
<i>Turn on bias</i>	0.1960 m/s^2	0.0087 rad/s
<i>Random Walk</i>	$0.002 \text{ m/s}^2/\sqrt{\text{s}}$	$0.0014 \text{ rad/s}/\sqrt{\text{s}}$

De seguida o sistema USBL, a simulação deste sensor foi realizada de forma diferente aos outros sensores. Para este foi definido uma *noise density* de $0.5 \text{ º}/\sqrt{\text{s}}$ nos dois ângulos obtidos pelo USBL, os ângulos *Yaw* e *Pitch*. Para a distância medida pelo sensor foi fixado uma *noise density* de $0.05 \text{ s}/\sqrt{\text{s}}$. Relativamente à taxa de aquisição de dados, aqui foi simulado a transmissão do sinal na água, sendo que quanto mais longe o *transponder* e o *pinger* menor será a taxa de aquisição dos dados.

6.2 Validação do filtro de Kalman estendido - localização doca

Nesta secção é apresentado a validação do filtro EKF para a localização da doca no seu referencial de navegação. Aqui a estrutura para cada um dos quatro trios de estados, posição, velocidade linear, orientação e velocidade angular, é apresentado uma comparação gráfica entre a estimada e a verdadeira acompanhada por uma tabela de quantificação do erro e, para além disso, é exposto graficamente a inovação para cada uma das observações, quando esta existe. Alguns testes suplementares

foram realizados e os seus resultados são também expostos, comparando estes com a estimação.

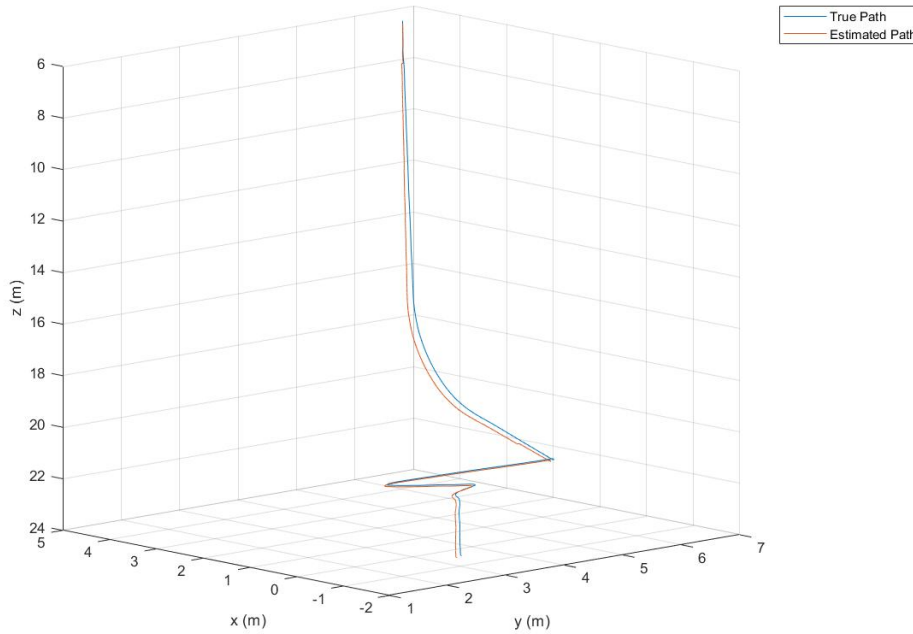


Figura 6.1: Comparação da trajetória estimada e a verdadeira

Primeiramente é apresentado, na figura 6.1, a comparação entre a trajetória estimada e a trajetória verdadeira.

De seguida, é analisado cada componente dessa trajetória. Ou seja, é comparado a estimação em posição x , y e z com a posição verdadeira em cada instante. Na figura 6.2, para além de ser apresentado a posição estimada e a verdadeira, é também exposto a informação obtida do USBL, em x e em y . Sendo possível verificar que estes dados são enviados em intervalos de tempo elevados quando comparado com a frequência a que a filtragem é realizada; que é de 50 Hz.

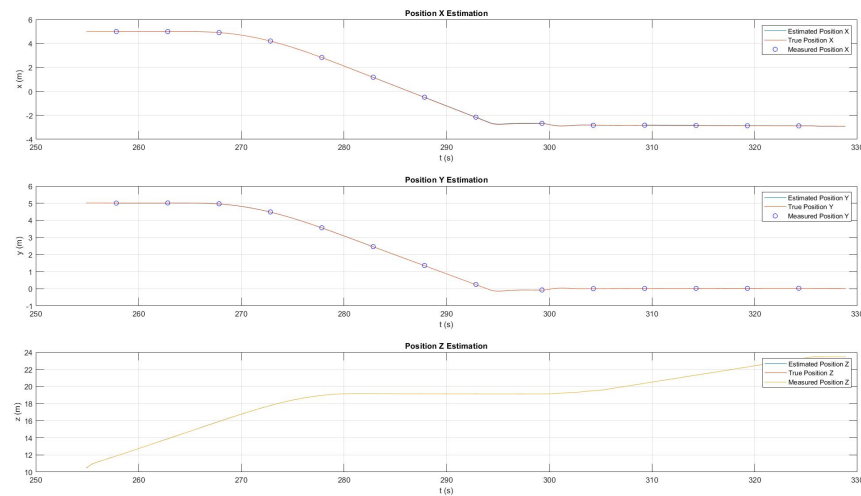


Figura 6.2: Comparação entre a posição estimada e a verdadeiras, em cada um dos três eixos

De forma a entender mais pormenorizadamente o erro da estimação, este é quantificado na tabela 6.2. Assim como, o desvio padrão da estimação em cada uma das componentes da posição.

Tabela 6.2: Quantificação do erro e desvio padrão em posição do sistema de localização do sistema de docagem

	em X	em Y	em Z
Erro máximo	0.1540 m	0.0756 m	0.1028 m
Erro médio	0.1052 m	0.0517 m	0.0204 m
Desvio Padrão	0.0416 m	0.0234 m	0.0295 m

Por fim, relativamente à estimação da posição, é apresentado a inovação relativa a esses estados, figura 6.3. Importa referir que a inovação traduz a diferença entre as observações e a previsão das observações, com base no estado previsto.

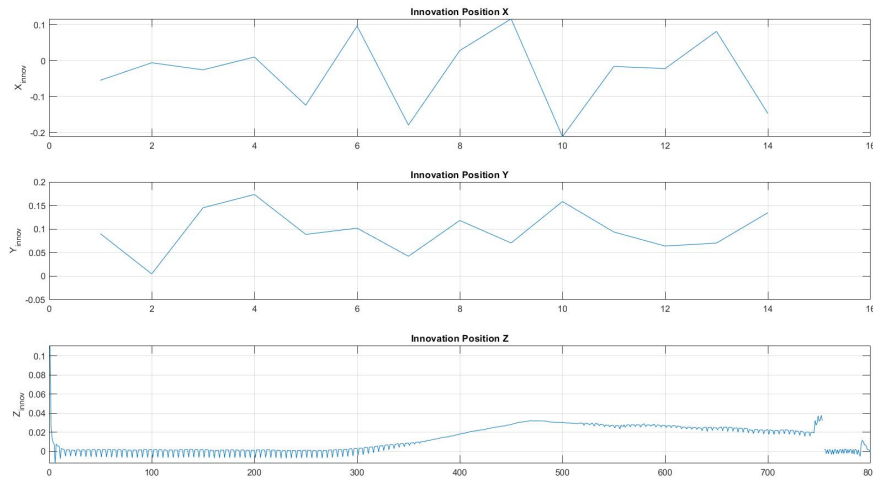


Figura 6.3: Inovação da posição em x , y e z

Com os resultados da posição apresentado, avançam-se para os seguintes estados, as componentes da velocidade linear da doca. Como referido estes estados são diretamente observados pelo DVL, sendo que, como estamos perante dados de uma simulação, efeitos derivados das condições ambientais não são tido em consideração.

A figura 6.4 é apresentado a comparação entre a estimação e a velocidade verdadeira do sistema de docagem. Aqui é possível perceber que a estimação destes estados é mais ruidosa, em comparação com a estimação da posição.

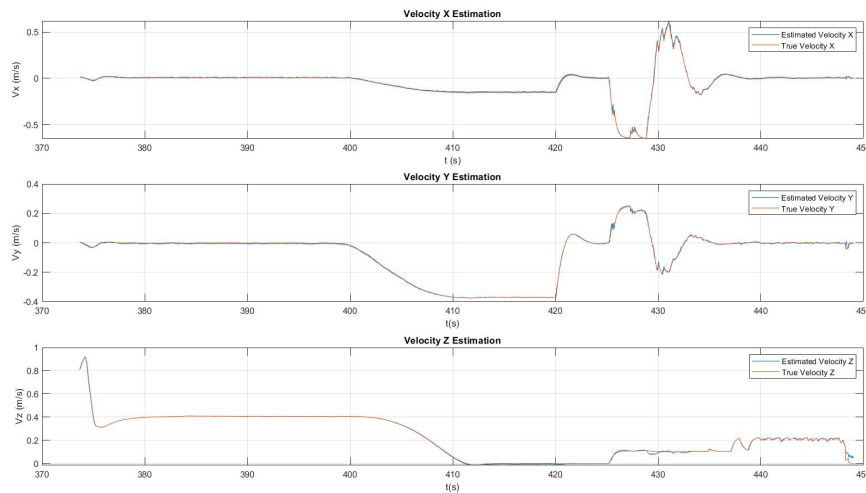


Figura 6.4: Comparação entre as velocidades estimadas e as verdadeiras, em cada um dos três eixos

Tal como na posição, achou-se por bem apresentar uma tabela com o erro, entre a estimação e as velocidades verdadeiras, de forma a perceber a precisão do filtro.

Estes dados são expostos na tabela 6.3.

Tabela 6.3: Quantificação do erro e desvio padrão em velocidade do sistema de localização do sistema de docagem

	em X	em Y	em Z
Erro máximo	0.1151 m/s	0.0809 m/s	0.1823 m/s
Erro médio	0.0046 m/s	0.0031 m/s	0.0022 m/s
Desvio Padrão	0.0110 m/s	0.0060 m/s	0.0095 m/s

Por fim, em relação às velocidades lineares, é apresentado a inovação referente à observação das mesmas, em 6.5.

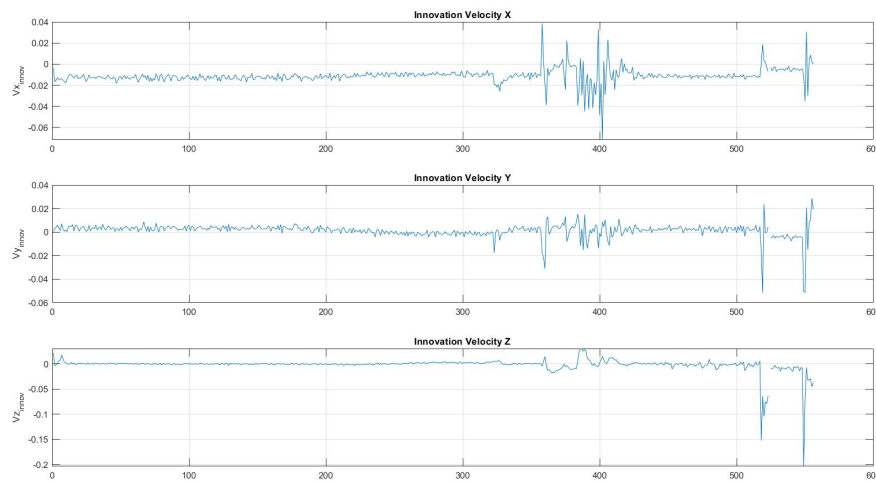


Figura 6.5: Inovação da velocidade em x , y e z

Por fim a orientação da doca em cada um dos eixos. A tabela, 6.4, da quantificação do erro volta a ser utilizada para apresentar quantificar o erro durante a estimação dos ângulos *roll*, *pitch* e *yaw*.

Tabela 6.4: Quantificação do erro e desvio padrão em orientação do sistema de localização do sistema de docagem

	em Roll	em Pitch	em Yaw
Erro máximo	6.3600 °	3.7738 °	10.0625 °
Erro médio	0.1002 °	0.2328 °	0.8104 °
Desvio Padrão	0.7258 °	0.3013 °	0.6806 °

Assim, na figura 6.6 é exibido uma comparação entre a orientação estimada e a verdadeira, em cada um dos eixos (ϕ , θ e ψ). Aqui o ruído da estimação é justificado pela observação direta destes ângulos. Uma vez que a informação obtida pelo sistema INS é também bastante ruidosa, mesmo em simulação.

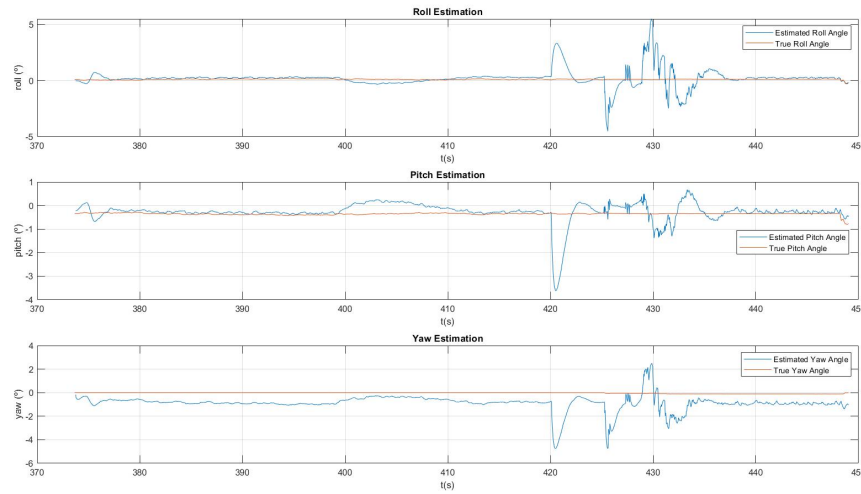


Figura 6.6: Comparação entre a orientação estimada e a verdadeira, em cada um dos três eixos

Tal como a posição e a velocidade linear, é também apresentado a inovação relativa à orientação. Esta é apresentada na figura 6.7.

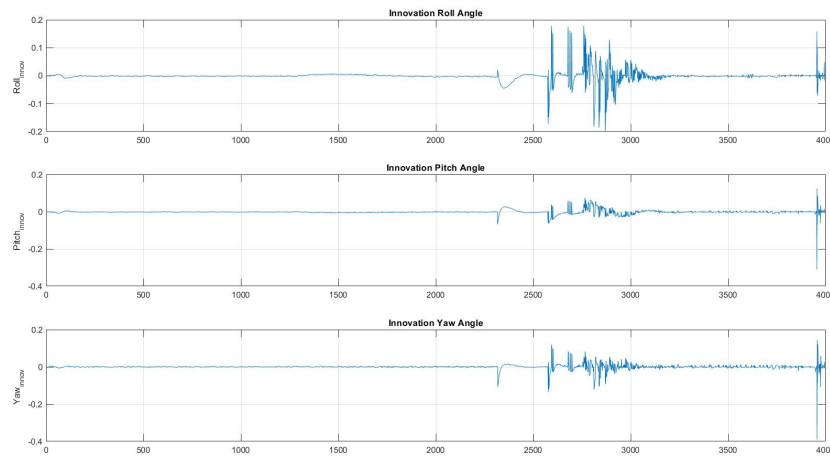


Figura 6.7: Inovação da orientação em ϕ , θ e ψ

Por não se observar diretamente a velocidade angular em cada um dos eixos, não é possível obter a inovação desses três estados. Ainda assim, e uma vez que se trabalha em contexto de simulação, é possível obter as componentes da velocidade angular verdadeira, figura 6.8.

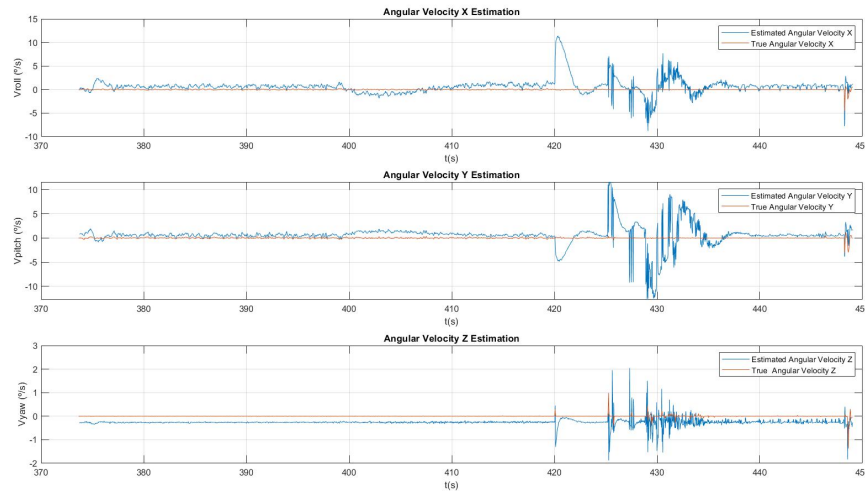


Figura 6.8: Comparação entre a velocidade angular estimada e a verdadeira, em cada um dos três eixos

Por isso, é apresentado na tabela 6.5 o erro e desvio padrão em velocidade angular.

Tabela 6.5: Quantificação do erro e desvio padrão em velocidade angular do sistema de localização do sistema de docagem

	em Roll	em Pitch	em Yaw
Erro máximo	6.4494 °/s	7.5029 °/s	11.3354 °/s
Erro médio	0.2134 °/s	0.1459 °/s	0.8335 °/s
Desvio Padrão	0.7527 °/s	0.4309 °/s	0.7069 °/s

6.2.1 Testes suplementar - Ruído em USBL e DVL

Claro que o facto de os dados trabalhados serem resultado de uma simulação, influencia a validação ou o grau de perfeição do filtro. De forma a testar a robustez do algoritmo, foi realizado um teste onde se eliminou algumas amostras do DVL e contaminar com algum ruído os dados proveniente do sistema USBL. Este teste simula as propensão ao ruído e a falhas dos sinais acústicos.

No DVL pretende-se simular as possíveis reflexão dos *beams* em algo que não seja o fundo do mar, levando a que esses dados não sejam utilizados. Assim, o que se realizou foi eliminar certos dados do DVL de uma forma aleatória, por volta dos 65% dos dados. No caso do USBL, a ideia é contaminar os dados deste com algum ruído, aumentando ainda mais o possível ruído que o sinal acústico poderá ter. De notar que este ruído foi adicionado após a obtenção dos dados e é realizado de forma aleatória. O valor de ruído adicionado varia entre -1.5 a 1.5 m.

Assim é possível verificar que a influência da perda de 65% dos dados relativos ao DVL e do aumento do ruído no USBL é mínima no resultado da estimação. Verificando assim a robustez do filtro probabilístico desenvolvido. Na tabela 6.6 é avaliado o erro e o desvio padrão dos nove estados, para o teste descrito.

Tabela 6.6: Quantificação do erro e desvio padrão dos estados para o teste referido

	Erro Máximo	Erro Médio	Desvio Padrão
Posição X	0.4520 m	0.1263 m	0.1131 m
Posição Y	0.4996 m	0.2067 m	0.1313 m
Posição Z	0.1067 m	0.0400 m	0.0205 m
Velocidade X	0.1635 m/s	0.0166 m/s	0.0251 m/s
Velocidade Y	0.0543 m/s	0.0052 m/s	0.0081 m/s
Velocidade Z	0.1186 m/s	0.0022 m/s	0.0104 m/s
Roll	5.4017 °	0.1501 °	0.6924 °
Pitch	3.2734 °	0.1236 °	0.3662 °
Yaw	4.7412 °	0.8157 °	0.5971 °
Velocidade Roll	5.4912 °/s	0.2474 °/s	0.6934 °/s
Velocidade Pitch	3.8271 °/s	0.2534 °/s	0.4181 °/s
Velocidade Yaw	4.7449 °/s	0.8461 °/s	0.6061 °/s

6.2.2 Testes suplementar - DVL a funcionar a uma certa profundidade

Um outro teste, realizado de forma a ver como o filtro se comportava, foi simular a obtenção das velocidades lineares, apenas a partir de uma certa profundidade. Neste foram eliminados os dados de quando o DVL se encontrava acima de 35 m relativamente ao fundo do mar. Aqui tenta-se simular uma possível situação onde o DVL não possuir alcance suficiente para os *beams* refletirem no fundo do mar. Os resultados encontram-se espelhados na tabela 6.7 e como é possível verificar, o erro em posição e em velocidade aumentam, como seria de esperar. Em velocidade, o erro médio duplicou em quase todas as suas componentes, passando a um erro de quase 0.01 m/s. Em posição, a falha no DVL provoca um aumento no erro máximo, na componente em x e em y . Este teste e os seus resultados contribuem para validar a robustez e a performance do filtro.

Tabela 6.7: Quantificação do erro e desvio padrão dos estados para o teste referido

	Erro Máximo	Erro Médio	Desvio Padrão
Posição X	4.7788 m	0.2632 m	1.2350 m
Posição Y	1.9525 m	0.0153 m	0.5237 m
Posição Z	0.1018 m	0.0208 m	0.0265 m
Velocidade X	0.2248 m/s	0.0088 m/s	0.0944 m/s
Velocidade Y	0.1011 m/s	0.0050 m/s	0.0384 m/s
Velocidade Z	0.1823 m/s	0.0023 m/s	0.0095 m/s
Roll	5.4107 °	0.1422 °	0.6807 °
Pitch	3.7738 °	0.1984 °	0.3646 °
Yaw	10.0625 °	0.8096 °	0.6796 °
Velocidade Roll	5.5002 °/s	0.2377 °/s	0.7188 °/s
Velocidade Pitch	7.5029 °/s	0.2066 °/s	0.4853 °/s
Velocidade Yaw	11.3354 °/s	0.8332 °/s	0.7060 °/s

6.2.3 Conclusão da validação do filtro

Com os resultados espelhados nesta secção é possível verificar que o filtro consegue estimar os estados com precisão, e por isso, corretamente. Relativamente à posição é possível verificar que o erro médio é superior nas componentes x e y , isto deve-se ao facto a informação do USBL ser um suscetível a mais ruído, quando comparando com o sensor de pressão. A velocidade linear, observada diretamente pelo DVL, possui erros médios na ordem dos 0.01 m/s . Para a orientação, o erro é superior em Yaw, quase na ordem de 1° , enquanto em Roll e Pitch, o mesmo ronda à volta dos 0.2° . Algo idêntico verifica-se nas velocidades angulares.

Todo o processamento do *rosbag* foi realizado no MATLAB com um processador AMD Ryzen 7 4800HS a 2.90 GHz. A tarefa de obtenção de todos os dados e realizar a estimação durou cerca de 55 segundos, sendo este $\frac{11}{16}$ do tempo de duração do *rosbag* (tempo total do bag é de aproximadamente 80 segundos) provando que é possível correr o algoritmo em tempo real. Ademais, importa ressaltar que a tarefa com maior custo computacional é a extração dos dados do *bag*, uma vez que a execução do filtro apenas dura 3.4 segundos, $\frac{17}{400}$ do tempo original. Garantindo, a certeza do funcionamento do filtro em tempo real.

6.3 Validação do filtro de Kalman estendido - localização doca $\leftrightarrow$ Turtle

Para a validação e verificação da robustez e fiabilidade do filtro relativo à localização do Turtle em relação à estação de docagem, são apresentados vários gráficos de comparação entre a estimação e o que realmente aconteceu na simulação.

Em contrário ao que se expôs na secção 6.2, aqui irá se apenas realizar uma comparação com a posição. Pois são estes três estados os mais importantes para a validação do filtro. Assim, na figura 6.9 é apresentado a comparação entre a posição relativa verdadeira e a estimada. Nesta é possível verificar que como existem poucos dados do sistema USBL, a estimação até à parte da visão é pobre.

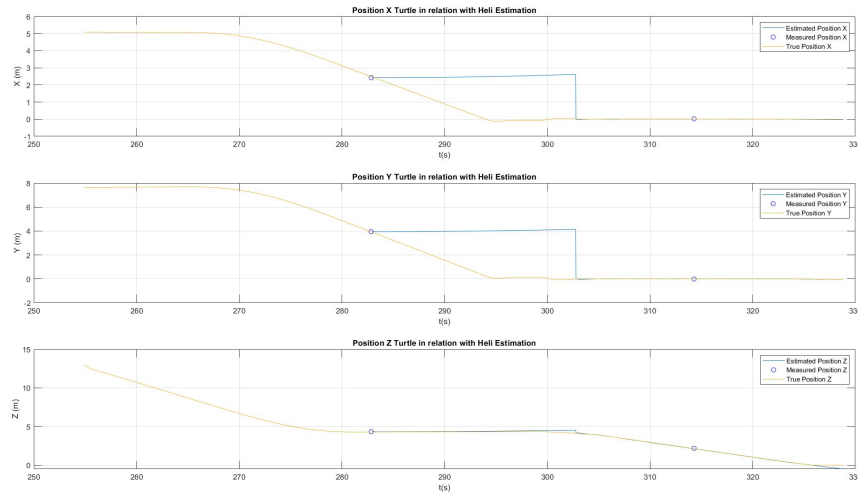


Figura 6.9: Comparação entre a posição estimada e a verdadeira, em cada um dos três eixos

Para a tabela relativa à quantificação do erro, tabela 6.8, a ideia foi a mesma da figura 6.10. Ou seja, só se considerou o momento em que o feedback da câmara é recebido no filtro. Claro que se levar em conta toda a filtragem, com os dados do USBL, o erro tende crescer quanto menos informação se obter desse sistema. No caso de existir mais informação, o filtro funcionará melhor.

Tabela 6.8: Quantificação do erro e desvio padrão da posição do sistema de localização entre o Turtle e a doca, após a primeira informação da câmara

	em X	em Y	em Z
Erro Máximo	0.0327 m	0.0549 m	0.1580 m
Erro Médio	0.0037 m	0.0062 m	0.0240 m
Desvio Padrão	0.0039 m	0.0072 m	0.0128 m

Pela razão apresentada atrás, é exposto uma comparação, figura 6.10 a partir do momento em que se obtém informação da câmara.

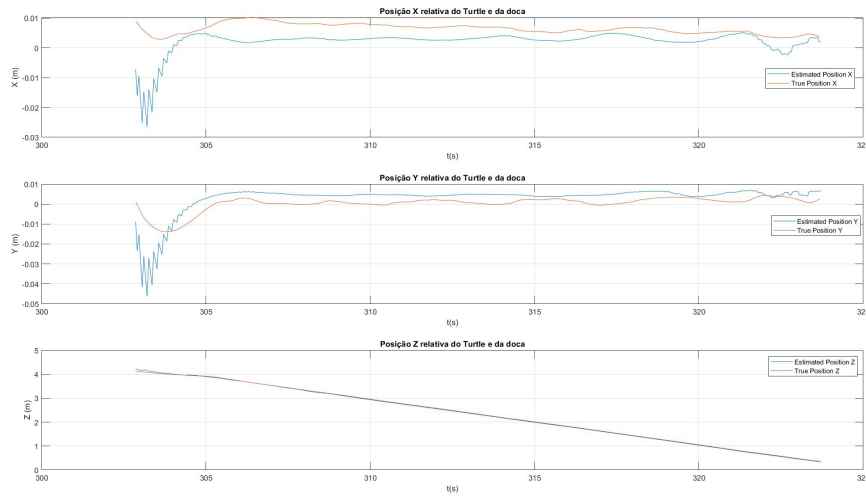


Figura 6.10: Comparação entre a posição estimada e a verdadeira, em cada um dos três eixos, em foco o momento após a primeira informação da câmara

A figura 6.11 apresenta a inovação da posição em cada um dos três eixos. Como seria de esperar, tendo em conta os gráficos apresentados acima, existe um pico no início provocado pelo intervalo de tempo entre a informação do USBL e a primeira informação da câmara.

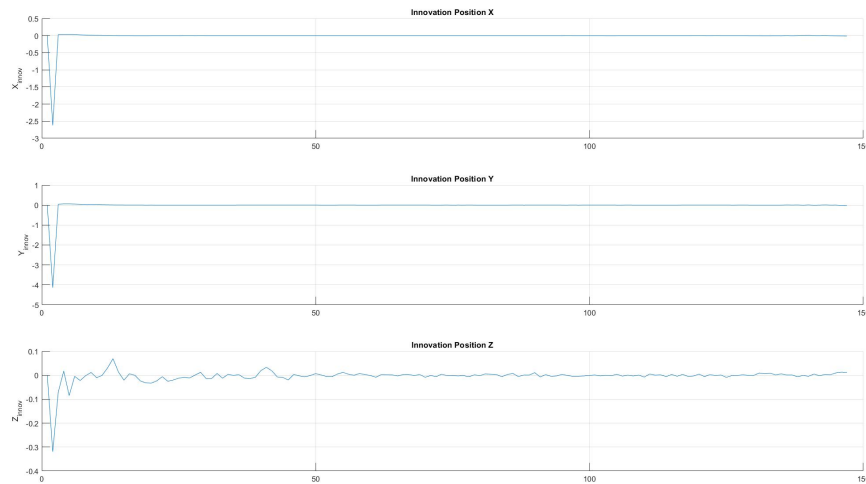


Figura 6.11: Inovação da posição em x , y e z

O facto de haver pouca informação proveniente do USBL promove o aumento do erro até ao momento que é recebido o feedback da câmara. Caso houvesse mais informação, a estimação nesse momento seria mais robusta. Olhando para após a câmara detetar o Turtle, verificamos que o erro anda na ordem dos mm, sendo que

a estimação em Z é claramente pior que a estimação nos restantes eixos.

Tal como no anterior a validação deste algoritmo foi realizada no MATLAB com um processador AMD Ryzen 7 4800HS a 2.90 GHz. Como seria de esperar, esta tarefa tem uma duração bastante elevada, em relação à anterior. Ao se adicionar ao sistema de localização o filtro EKF doca $\leftrightarrow$ Turtle, o tempo de execução passa para perto dos 43 segundos, metade do tempo de duração do *roslab*. Este aumento é expectável, uma vez que é neste filtro que é utilizado o algoritmo mais pesado computacionalmente, de todo o sistema de localização, o sistema de visão. Mesmo assim, é de destacar que o tempo de execução inicial se encontrava na ordem dos 150 segundos, sendo que se reduziu a frequência do sistema de visão. Deixando de utilizar menos *frames* para processamento mas não aumentando o erro do sistema de localização. Com este tempo de execução, de 43 segundos, prova que é possível correr o algoritmo em tempo real, visto ser metade do tempo de duração do *roslab*.

6.4 Validação do sistema de visão

De forma a validar o sistema de visão desenvolvido são expostas várias imagens retiradas diretamente da câmara simulada, sendo que é mostrado o que cada filtro faz à imagem e é também apresentada a eficácia do sistema de deteção do centro do Turtle.

Na figura 6.12 são apresentados os diferentes passos realizados no algoritmo de visão, sendo estes os seguintes:

- A imagem *raw* obtida diretamente da câmara;
- Aplicação do filtro preto e branco;
- Resultado do filtro binário.

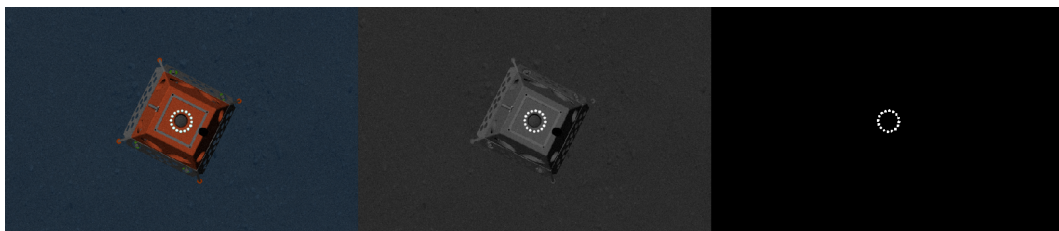


Figura 6.12: Passos realizados no algoritmo de visão: Imagem retirada da câmara, aplicação preto e branco e aplicação do *threshold*

A imagem com que se obtém o centro do Turtle é o último *frame* da sequência apresentada na figura 6.12.

Para a verificação do algoritmo de visão, são apresentados quatro exemplos. Nestes é possível comparar o centro real do Turtle com o centro obtido pelo algoritmo de visão. A figura 6.13 mostra esses quatro exemplos, nestes é também apresentado,

no canto superior, a posição x , y e z do turtle em relação ao sistema de docagem. Para além disso, as imagens são divididas em quatro quadrantes de forma a perceber se o Turtle se encontra ou não centrado.

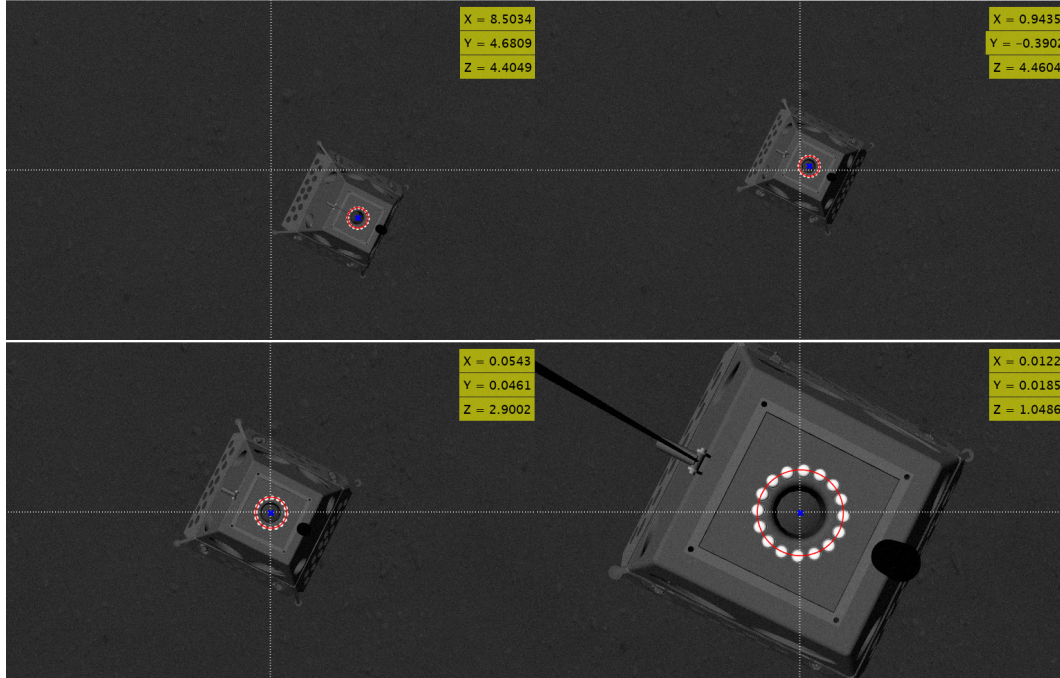


Figura 6.13: Vários exemplos do output do sistema de visão

Por último, uma comparação interessante é perceber o quão preciso é o cálculo da posição quando comparado com a posição relativa verdadeira. Esta comparação é possível de ser feita, uma vez que no simulador consegue saber a posição de cada um dos sistemas. A tabela 6.9 apresenta o erro máximo entre a posição verdadeira e a obtida pelo algoritmo desenvolvido.

Tabela 6.9: Erro máximo entre a posição verdade e obtida pelo algoritmo de visão

	em X	em Y	em Z
Erro máximo	0.0065 m	0.0090 m	0.0681 m

Aqui optou-se por mostrar o erro máximo para perceber se o algoritmo de visão consegue estar sempre relativamente perto do valor real.

6.5 Comparação entre filtro do modelo dinâmico com filtro INS

Um dos testes interessantes a realizar é comparar o filtro que utiliza o modelo dinâmico na fase de previsão com um outro filtro denominado filtro INS. Neste os

dados do sistema INS (velocidade angular e aceleração linear) são utilizados na fase da previsão. Para isso foi importante desenvolver um filtro EKF diferente do já apresentado. Neste a fase de previsão faz uso das acelerações lineares e velocidades angulares do sistema INS. A fase de atualização é idêntica à apresentada em 5.1.2, apenas não observado a orientação diretamente.

Com isto, na tabela 6.10 é possível comparar o erro médio, de cada um dos doze estados, dos dois filtros. Como seria de esperar o filtro, com base no sistema INS, tende a ter um erro superior ao EKF, com base no modelo dinâmico. Isto verifica-se mais na estimação da posição e velocidade em Z, assim como em roll e pitch e respetivas velocidades angulares.

Tabela 6.10: Quantificação do erro médio para os doze estados nos dois filtros diferentes

	EKF INS	EKF modelo dinâmico
Posição em x	0.1382 m	0.1070 m
Posição em y	0.1488 m	0.0504 m
Posição em z	0.0695 m	0.0349 m
Velocidade em x	0.0357 m/s	0.0048 m/s
Velocidade em y	0.0161 m/s	0.0032 m/s
Velocidade em z	0.6299 m/s	0.0019 m/s
Roll	8.9716 °	0.1551 °
Pitch	8.5306 °	0.1162 °
Yaw	8.1785 °	0.8157 °
Velocidade em roll	9.0837 °/s	0.2523 °/s
Velocidade em pitch	8.1836 °/s	0.2634 °/s
Velocidade em yaw	8.1681 °/s	0.8461 °/s

Onde se verifica maiores discrepâncias na comparação entre os dois filtros, é quando se adiciona as falhas no DVL e ruído aos dados no sistema USBL. Neste teste, diferente dos apresentados anteriormente, é adicionado ruído a todas as amostras do USBL e o DVL só funciona a uma certa profundidade, 35 m. Com isto, o erro médio de cada estado é apresentado na tabela 6.11.

Apesar de neste teste, o erro médio ser maior no EKF, com base no modelo dinâmico, do que nos outros testes, quando comparado com o filtro, com base no sistema INS, o erro ainda é menor. Garantindo assim que o filtro do modelo dinâmico é mais fiável e preciso que a outra estimação. Validando assim a utilização de um modelo dinâmico para a fase de previsão.

Tabela 6.11: Quantificação do erro médio para os doze estados nos dois filtros diferentes com falhas no DVL e USBL

	EKF INS	EKF modelo dinâmico
Posição em x	2.7896 m	0.4059 m
Posição em y	0.9188 m	0.7488 m
Posição em z	0.0701 m	0.0454 m
Velocidade em x	0.4447 m/s	0.0715 m/s
Velocidade em y	0.1494 m/s	0.0116 m/s
Velocidade em z	0.6394 m/s	0.0037 m/s
Roll	10.4430 °	0.1353 °
Pitch	9.9175 °	0.1640 °
Yaw	9.6443 °	0.8161 °
Velocidade em roll	10.5549 °/s	0.2288 °/s
Velocidade em pitch	9.5753 °/s	0.2341 °/s
Velocidade em yaw	9.5600 °/s	0.8461 °/s

Capítulo 7

Conclusões

Esta dissertação pretendeu desenvolver um sistema de localização para a nova estação de docagem. Esta terá que docar no lander robótico Turtle de forma a carregá-lo. Uma vez que é totalmente autónomo, um sistema de localização capaz de localizar a doca no seu referencial de navegação e o Turtle no referencial da doca é necessário.

Apesar de não ser o foco desta dissertação, a apresentação dos sensores serve como indicação para o tipo de sistemas sensoriais que estarão presentes no ambiente da missão; tanto como no Turtle, como no sistema de docagem.

A componente mais importante desta dissertação são os dois sistemas de estimação. Um para a localização da doca no seu referencial de navegação e outro para localizar o Turtle em relação à doca.

O filtro modelo dinâmico mostrou-se robusto à falha de medidas sensoriais, como exposto nos resultados. A simulação de falhas no DVL e o aumento do ruído nas amostras do USBL provam esse ponto. No funcionamento sem estas falhas, o erro médio em posição é inferior a 10 cm, durante a execução da manobra de docagem. Sendo que relativamente à orientação o erro médio é inferior a 1°. A eficiência do filtro a estimar as doze variáveis de estado é garantido pelo modelo dinâmico utilizado como aproximação do sistema de docagem.

A secção 6.5 mostra como a utilização do modelo dinâmico para a fase de previsão é fundamental para o bom funcionamento da filtragem. Quando comparando os dois filtros, verifica-se que o erro em orientação do EKF INS ronda os 10°, enquanto no filtro do modelo dinâmico, ronda os 1°. Na estimação para a posição e para a velocidade linear a diferença já não é tanta como na orientação. Quando se adiciona

ruído aos dados do USBL e o DVL só funciona a uma certa profundidade, verifica-se que o erro no filtro INS ainda é maior quando comparando com o filtro do modelo dinâmico. O filtro com base no INS beneficiaria de uma possível calibração do sistema INS utilizado.

Relativamente à estimação da localização relativa provou-se eficaz, apesar da pouca informação proveniente do sistema USBL. Neste filtro os estados estimados, que interessam para o sistema de localização, são a posição em cada os dos seus eixos. O facto de haver pouca informação por parte do USBL torna os resultados pobres, até ao momento da utilização da câmara. Isto poderia ser facilmente contrariado com o aumento de transmissão de dados entre o Turtle e a bóia.

O sistema de localização Doca $\leftrightarrow$ Turtle desenvolvido demorou cerca de metade do tempo do *rosbag* para processar e estimar os estados com um processador AMD Ryzen 7 4800HS a 2.90 GHz; demonstrando assim que é possível o funcionamento em tempo real. Como ficou mostrado, o tempo de execução aumenta bastante por causa do processamento de imagem importante para detetar o Turtle e a distância da doca a esta. No sistema de localização da doca no seu referencial de navegação, o tempo de execução é de 3.4 segundos, $\frac{17}{400}$ do tempo total do *bag*. Em suma, os objetivos propostos inicialmente para esta dissertação foram cumpridos com sucesso.

7.1 Trabalho Futuro

Relativamente às possíveis melhorias no futuro, desenvolver um processo de calibração para sensores como o INS. Este processo poderá ajudar a reduzir alguns erros associados a este sensor.

Uma possível melhoria poderia ser alterar o algoritmo de visão para algo mais complexo, como reconstrução 3D através de um sistema de visão *stereo*. Esta abordagem poderia melhorar a precisão da estimação da profundidade.

O modelo dinâmico da doca é uma aproximação do modelo real, uma melhoria passaria por realizar todos os passos de forma a estimar com maior precisão o modelo dinâmico.

Referências

- [1] L. Paull, S. Saeedi, M. Seto, and H. Li, “AUV navigation and localization: A review,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014. [Citado nas páginas vii, 10 e 15]
- [2] K. M. Alzahrani, “An Underwater Vehicle Navigation System Using Acoustic and Inertial Sensors,” 2018. [Citado nas páginas vii, 10, 12 e 13]
- [3] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*. USA: McGraw-Hill, Inc., 1 ed., 2008. [Citado nas páginas vii, 17 e 18]
- [4] J. Farrell, “Aided navigation: Gps with high rate sensors,” 2008. [Citado nas páginas vii e 19]
- [5] L. G. García-Valdovinos, T. Salgado-Jiménez, M. Bandala-Sánchez, L. Nava-Balazar, R. Hernández-Alvarado, and J. A. Cruz-Ledesma, “Modelling, Design and Robust Control of a Remotely Operated Underwater Vehicle,” *International Journal of Advanced Robotic Systems*, vol. 11, no. 1, 2014. [Citado nas páginas vii e 19]
- [6] H. Gjoreski and M. Gams, “Activity/posture recognition using wearable sensors placed on different body locations,” *Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing, ASC 2011*, no. December 2013, pp. 340–347, 2011. [Citado nas páginas vii e 20]
- [7] EvoLogics, “S2C T 18/34 | EvoLogics.” Available at <https://evologics.de/acoustic-modem/18-34/t-series>. (Last accessed in 17/09/2021). [Citado nas páginas vii e 32]
- [8] W. L. AS, “DVL A50 - Water Linked AS.” Available at <https://store.waterlinked.com/product/dvl-a50/>. (Last accessed in 17/09/2021). [Citado nas páginas vii e 33]
- [9] A. D. Team, “The Cube Orange With ADBS-In Overview - Copter documentation.” Available at <https://ardupilot.org/copter/docs/common-thecubeorange-overview.html>. (Last accessed in 17/09/2021). [Citado nas páginas vii e 33]

-
- [10] U. G. Survey, “How Much Water is There on Earth?.” Available at https://www.usgs.gov/special-topic/water-science-school/science/how-much-water-there-earth?qt-science_center_objects=0#qt-science_center_objects. (Last accessed in 02/09/2021). [Citado na página 1]
- [11] B. L. Matias, “Sistema de Localização Subaquático para Veículos Autônomos,” 2016. [Citado nas páginas 1, 9, 12 e 18]
- [12] S. Petillo and H. Schmidt, “Exploiting adaptive and collaborative AUV autonomy for detection and characterization of internal waves,” *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 150–164, 2014. [Citado na página 1]
- [13] P. Ridao, M. Carreras, D. Ribas, P. J. Sanz, and G. Oliver, “Intervention AUVs: The next challenge,” *Annual Reviews in Control*, vol. 40, pp. 227–241, jan 2015. [Citado na página 2]
- [14] N. Abreu and A. Matos, “Minehunting Mission Planning for Autonomous Underwater Systems Using Evolutionary Algorithms,” *Unmanned Systems*, vol. 02, no. 04, pp. 323–349, 2014. [Citado na página 2]
- [15] H. Ferreira, A. Martins, J. M. Almeida, A. Valente, A. Figueiredo, B. Da Cruz, M. Camilo, V. Lobo, C. Pinho, A. Olivier, E. Silva, and S. Matos - Metalomecanica, “TURTLE - Systems and technologies for Deep Ocean long term presence,” [Citado na página 2]
- [16] S. Korpilo, T. Virtanen, and S. Lehvävirta, “Smartphone GPS tracking—Inexpensive and efficient data collection on recreational movement,” *Landscape and Urban Planning*, vol. 157, pp. 608–617, 2017. [Citado na página 7]
- [17] Tesla, “Tesla Self-Driving Demonstration | Tesla Europe.” Available at https://www.tesla.com/en_EU/videos/autopilot-self-driving-hardware-neighborhood-short?redirect=no. (Last accessed in 02/09/2021). [Citado na página 7]
- [18] M. B. Alatise and G. P. Hancke, “A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods,” *IEEE Access*, vol. 8, pp. 39830–39846, 2020. [Citado na página 8]
- [19] S. Holtzhausen, O. Matsebe, N. S. Tlale, and G. Bright, “Autonomous underwater vehicle motion tracking using a Kalman filter for sensor fusion,” *15th International Conference on Mechatronics and Machine Vision in Practice, M2VIP’08*, pp. 103–108, 2008. [Citado na página 8]

- [20] N. Y. Ko and T. Y. Kuc, “Fusing range measurements from ultrasonic beacons and a laser range finder for localization of a mobile robot,” *Sensors (Switzerland)*, vol. 15, no. 5, pp. 11050–11075, 2015. [Citado na página 8]
- [21] Y. Petillot, F. Maurelli, N. Valeyrie, A. Mallios, P. Ridao, J. Aulinas, and J. Salvi, “Acoustic-based techniques for autonomous underwater vehicle localization,” *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, vol. 224, no. 4, pp. 293–307, 2010. [Citado na página 9]
- [22] J. González-García, A. Gómez-Espinosa, E. Cuan-Urquizo, L. G. García-Valdovinos, T. Salgado-Jiménez, and J. A. Escobedo Cabello, “Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions,” *Applied Sciences (Switzerland)*, vol. 10, no. 4, 2020. [Citado na página 10]
- [23] L. Stutters, H. Liu, C. Tiltman, and D. J. Brown, “Navigation technologies for autonomous underwater vehicles,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 38, no. 4, pp. 581–589, 2008. [Citado na página 11]
- [24] R. Panish and M. Taylor, “Achieving high navigation accuracy using inertial navigation systems in autonomous underwater vehicles,” *OCEANS 2011 IEEE - Spain*, 2011. [Citado na página 11]
- [25] D. Ribas, P. Ridao, and J. Neira, *Localization with an a priori Map*, pp. 47–75. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. [Citado na página 12]
- [26] R. M. Eustice, O. Pizarro, and H. Singh, “Visually augmented navigation for autonomous underwater vehicles,” *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 103–122, 2008. [Citado na página 12]
- [27] D. Loebis, R. Sutton, and J. Chudley, “Review of multisensor data fusion techniques and their application to autonomous underwater vehicle navigation,” *Journal of Marine Engineering and Technology*, vol. 1, no. 1, pp. 3–14, 2002. [Citado na página 12]
- [28] D. Ribas, P. Ridao, A. Mallios, and N. Palomeras, “Delayed state information filter for USBL-Aided AUV navigation,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4898–4903, 2012. [Citado na página 14]
- [29] R. L. Roberts, R. G. Knapp, A. J. Healey, and M. J. Zyda, “System for Small AUV Navigation,” vol. 24, no. 3, pp. 396–404, 1999. [Citado na página 15]

-
- [30] W. Wang, “Autonomous Control of Differential Thrust Micro ROV,” no. March, 2006. [Citado na página 15]
- [31] E. A. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” *IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, AS-SPCC 2000*, pp. 153–158, 2000. [Citado na página 15]
- [32] E. A. Wan, R. Van Der Merwe, and A. T. Nelson, “Dual estimation and the unscented transformation,” *Advances in Neural Information Processing Systems*, pp. 667–672, 2000. [Citado na página 15]
- [33] P. Rigby, O. Pizarro, and S. B. Williams, “Towards geo-referenced AUV navigation through fusion of USBL and DVL measurements,” *Oceans 2006*, no. 2, 2006. [Citado na página 15]
- [34] S. I. Roumeliotis and G. A. Bekey, “Distributed multirobot localization,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002. [Citado na página 15]
- [35] G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis, “Observability-based consistent EKF estimators for multi-robot cooperative localization,” *Autonomous Robots*, vol. 30, no. 1, pp. 99–122, 2011. [Citado na página 15]
- [36] X. S. Zhou and S. I. Roumeliotis, “Robot-to-robot relative pose estimation from range measurements,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1379–1393, 2008. [Citado na página 15]
- [37] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, “Distributed maximum a posteriori estimation for multi-robot cooperative localization,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1402–1409, 2009. [Citado na página 15]
- [38] R. Gao and M. Chitre, “Cooperative positioning using range-only measurements between two AUVs,” *OCEANS’10 IEEE Sydney, OCEANSSYD 2010*, 2010. [Citado na página 15]
- [39] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus, “Experiments with cooperative control of underwater robots,” *International Journal of Robotics Research*, vol. 28, no. 6, pp. 815–833, 2009. [Citado na página 16]
- [40] D. K. Maczka, A. S. Gadre, and D. J. Stilwell, “Implementation of a cooperative navigation algorithm on a platoon of autonomous underwater vehicles,” *Oceans Conference Record (IEEE)*, 2007. [Citado na página 16]

-
- [41] M. F. Fallon, G. Papadopoulos, and J. J. Leonard, “A measurement distribution framework for cooperative navigation using multiple AUVs,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4256–4263, 2010. [Citado na página 16]
- [42] V. Nav, “Reference frame and how they used in inertial navigation Vector Nav.” Available at <https://www.vectornav.com/resources/inertial-navigation-primer/math-fundamentals/math-refframes>. (Last accessed in 10/09/2021). [Citado nas páginas 18 e 19]
- [43] L. Euler, “Formvlae generales pro translatione qvacvnqvve corporvm rigidor,” 1776. [Citado na página 20]
- [44] R. Academy, “Singularity in 3D rotation angles sequences | Robot Academy.” Available at <https://robotacademy.net.au/lesson/singularity-in-3d-rotation-angle-sequences/>. (Last accessed in 10/09/2021). [Citado na página 21]
- [45] A. L. Noonan, “FLIGHT PLAN GENERATION FOR UNMANNED AERIAL by Department of Mechanical and Nuclear Engineering,” 2007. [Citado na página 23]
- [46] D. M. Demers and R. Pinckney, “Population Risk Management and Workplace Health Centers,” *Population*, 2014. [Citado na página 23]
- [47] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. 2011. [Citado nas páginas 24, 25, 26 e 27]
- [48] V. Berg, “Development and Commissioning of a DP system for ROV SF 30k,” no. June, p. 98, 2012. [Citado na página 28]
- [49] EvoLogics, “S2C T 18/34D | EvoLogics.” Available at <https://evologics.de/acoustic-modem/18-34d/usbl-serie>. (Last accessed in 18/09/2021). [Citado na página 35]
- [50] Google, “Camera | Coral.” Available at <https://coral.ai/products/camera/>. (Last accessed in 18/09/2021). [Citado na página 35]
- [51] R. P. Foundation, “Getting started with the Camera Module - Introduction | Raspberry Pi Projects.” Available at <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera>. (Last accessed in 18/09/2021). [Citado na página 35]
- [52] S. J. Ahn, W. Rauh, and H. J. Warnecke, “Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola,” *Pattern Recognition*, vol. 34, no. 12, pp. 2283–2303, 2001. [Citado na página 49]

- [53] T. U. S. Authors, “Unmanned Underwater Vehicle Simulator Documentation.” Available at <https://uuvsimulator.github.io/>. (Last accessed in 25/09/2021). [Citado na página 51]