

Instituto Superior de Engenharia do Porto



Coordenação de veículo de inspecção subaquática com veículo de superfície autónomo

Pedro Manuel Oliveira Pereira

Tese submetida no âmbito do
Mestrado em Engenharia Electrotécnica e de Computadores

Orientador: Alfredo Manuel Oliveira Martins (Mestre)

Julho de 2011

Resumo

O LSA/ISEP(Laboratório de sistemas Autónomos do Instituto Superior de Engenharia do Porto) tem vindo nos últimos anos a desenvolver sistemas robóticos inovadores para operação em ambiente marinho sendo o veículo de superfície autónomo ROAZ II um exemplo de renome internacional. Neste contexto, e tendo em vista a satisfação dos requisitos parciais conducentes à obtenção do grau de Mestre em Eng. Electrotécnica e de Computadores - Ramo de Sistemas Autónomos do ISEP, o presente trabalho visou a integração de um robô submarino operado remotamente (ROV) com o robô de superfície ROAZ II. Esta solução inovadora de operação coordenada e integrada de um ASV/ROV permite dotar o ASV de mobilidade e visão subaquática. Após a caracterização e análise de requisitos de diversos cenários operacionais foi apresentada uma arquitectura de controlo coordenado dos dois veículos baseada em manobras de controlo descritas por autómatos híbridos. Os dois veículos foram modelados e as manobras coordenadas projectadas foram validadas com um simulador em ambiente *Matlab/Simulink*. Foi desenvolvido um sistema de localização relativa do ROV através da fusão sensorial de um sistema INS com um sistema acústico USBL utilizando um filtro EKF. O veículo ROV (VideoRay) do LSA foi instrumentado com os sensores necessários e efectuada a integração de hardware e software com o ASV ROAZ II permitindo a operação remota. Foi realizada uma missão demonstrativa de inspecção de pilares subaquáticos em cenário real com a operação conjunta dos dois robôs.

Palavras Chave:ROV, ASV, INS, USBL, KALMAN, Movimento Coordenado

Esta página foi intencionalmente deixada em branco.

Abstract

The LSA/ISEP has been developing innovative robotic systems for marine environment operations, the ROAZ II is an example of international reputation. In this scope, and taking care the satisfaction of the partial requirements to obtain a Master Degree in Eng. Electrotécnica e de Computadores - Ramo de Sistemas Autónomos, this work focus on the integration of a underwater remotely operated vehicle with a surface vehicle ROAZ II. This innovative operating solution in a coordinated and integrated ASV/ROV allows the ASV to have mobility and underwater vision. After the characterization and requirement of the multiple in operational scenario, a coordinated control architecture based on a hybrid automata has been presented. The two vehicle were modeled a coordinated maneuvers were projected and validated using a simulator in Matlab/Simulink environment. A relative localization system has been developed in each the data fusion of the INS and USBL is an EKF. The VideoRay vehicle was instrumented with the necessary equipment in order to integrate the Hardware and software with the ROAZ II. We performed a demonstration mission for inspection of underwater pillars in real scenario for the joint operation of two robots.

Keywords: ROV, ASV, INS, USBL, KALMAN, Coordinated Movement

Esta página foi intencionalmente deixada em branco.

Agradecimentos

Diz um provérbio chinês "A viagem é mais importante que o destino" e esta viagem de construção desta dissertação foi particularmente acidentada. Estou convicto que este é o melhor local para agradecer a todos aqueles que estiveram envolvidos nesta viagem.

Começo por agradecer ao meu orientador Eng. Alfredo Martins por todo o apoio e dedicação que demonstrou ao longo de toda a dissertação. Quero também agradecer ao Eng. Eduardo Silva e Eng. José Miguel Almeida pela ajuda que nunca recusaram em todo este processo. Gostaria de deixar uma palavra de carinho ao Eng. Hugo Ferreira que foi desde o início um grande mentor.

Gostaria de agradecer de forma especial aos meus companheiros Eng. Sérgio, Eng. Ivo, Eng. Tiago, Eng. João, Eng. Guilherme e Eng. António pelo facto de estarem sempre disponíveis para ajudarem, sem eles não teria sido possível realizar todas as missões em cenário operacional.

Gostaria de agradecer aos meus grandes amigos Pedro e Xana que estiveram presentes nos momentos mais difíceis desta viagem e por todo o carinho, preocupação e palavras de incentivo que tiveram.

Quero agradecer de uma forma muito especial aos meus pais que fizeram todos os esforços para que estivessem sempre reunidas as melhores condições possíveis não só no processo de elaboração da dissertação mas em toda a minha vida académica, a eles todo o meu carinho.

"*The last but not the least*" um agradecimento muito especial à minha futura esposa Andreia pela compreensão, ajuda, motivação e por todas as horas que prescindiu de estar comigo para que este processo tivesse o melhor final possível e sobretudo por ser a companheira de todas as horas. Sem ela tudo seria mais difícil.

Pedro Manuel Oliveira Pereira

Esta página foi intencionalmente deixada em branco.

Dedicatória

... À Andreia, aos meus Pais, ao Avô Zé

Esta página foi intencionalmente deixada em branco.

”Engineering is not merely knowing and being knowledgeable, like a walking encyclopedia; engineering is not merely analysis; engineering is not merely the possession of the capacity to get elegant solutions to non-existent engineering problems; engineering is practicing the art of the organized forcing of technological change... Engineers operate at the interface between science and society..”

Dean Gordon Brown

Esta página foi intencionalmente deixada em branco.

Conteúdo

1	Introdução	1
1.1	Âmbito da Dissertação	1
1.2	Objectivos e Contribuições	2
1.3	Organização da Tese	3
2	Problema de Coordenação entre ASV e ROV	5
2.1	Formulação do Problema	5
2.2	Cenários Operacionais De Coordenação ASV-ROV	8
2.2.1	Inspeção Teleoperada De Estruturas	8
2.2.2	Inspeção De Estruturas Extensas	9
2.2.3	Arqueologia Marinha	9
2.3	ROAZ	9
2.4	Sistemas Robóticos Submarinos do LSA-ISEP	10
3	Estado da Arte	13
3.1	<i>Autonomous Surface Vehicle</i>	13
3.2	<i>Remotely Operated Vehicle</i>	14
3.3	Múltiplos Veículos Robóticos Em Ambiente Marinho	14
3.3.1	Respiradouros Hidrotermais	16
3.3.2	Descarga De Dados De Peixes	16
3.3.3	Mapeamento De <i>Habitat</i>	18
3.4	Coordenação ASV AUV	18
3.5	Simuladores	19
3.5.0.1	<i>Micro Rov Simulator</i>	19
3.5.1	RDA (ROV Design And Analysis Toolbox)	20
3.5.2	Simuladores Comerciais	20
3.5.2.1	ROVsim	21
3.6	Navegação Subaquática	22
4	Fundamentos de Navegação	25
4.1	Técnicas De Navegação	25
4.1.1	Sistemas Baseados Em Pontos De Referência	25
4.1.2	Sistemas " <i>Dead Reckoning</i> "	26
4.1.3	Sistemas de Navegação Inercial (INS)	26
4.2	Sistemas de Coordenadas	28
4.2.1	Referenciais	28
4.3	Notação Usada	29
4.3.1	Cinemática	31

5	Modelação	33
5.1	Modelo Dinâmico	33
5.2	Dinâmica de um corpo rígido	33
5.3	Forças Hidrodinâmicas	35
5.3.1	Massa Acrescentada	35
5.3.2	Amortecimento Hidrodinâmico	36
5.3.3	Forças Hidrostática	37
5.4	Equações Do Movimento	38
5.4.1	Modelo Reduzido	38
5.5	Modelo De Um Robô Diferencial	39
6	Projecto	43
6.1	Arquitectura Do Sistema	43
6.2	Sistema Auxiliar De Instrumentação	45
6.2.1	Sistema Computacional	45
6.2.2	Controlo De Energia	45
6.2.3	Hardware	46
6.3	INS	47
6.4	Arquitectura Do Veículo De Superfície Autónomo ROAZ II	47
6.5	Controlo	48
6.5.1	Controlo De Trajectória	48
6.5.1.1	Orientação em linha recta	49
6.5.1.2	Orientação em curva	49
6.6	Coordenação entre ASV e ROV	51
6.7	Navegação USBL	52
6.7.1	Sistema USBL <i>Tritech</i>	52
6.8	Autómato Híbrido	54
6.8.1	<i>StateFlow</i>	55
6.8.2	Exemplo <i>StateFlow</i>	57
6.9	Estimação e filtro de <i>Kalman</i>	59
6.9.1	Algoritmo De Filtro De <i>Kalman</i>	61
6.9.2	Filtro De <i>Kalman</i> Não Linear	62
6.9.2.1	<i>EKF - Extended Kalman Filter</i>	64
7	Implementação e Resultados	67
7.1	Integração do sistema USBL	67
7.1.1	Hardware	67
7.1.2	Software	67
7.1.3	Testes Preliminares	68
7.2	Sistema Auxiliar de Instrumentação	70
7.2.1	Interligações	71
7.3	INS	71
7.4	Simulador	71
7.5	Resultados Do Simulador	76
7.5.1	Teleoperação	76
7.5.2	Funcionamento Autónomo	77
7.5.3	Condições Da Simulação	79
7.6	Implementação De Filtro De Localização	80
7.7	Integração Preliminar do ASV ROV	83

7.7.1	Cenário Operacional	86
8	Conclusões e Trabalho Futuro	91
8.1	Conclusões	91
8.2	Trabalho Futuro	92
	Referências	96

Esta página foi intencionalmente deixada em branco.

Lista de Figuras

2.1	cenário operacional	7
2.2	Veículo ROAZ II	10
2.3	<i>VideoRay</i> PRO III	11
2.4	<i>SeaBotix</i> LBV150	11
3.1	Representação 2D de VBAP	15
3.2	Busca de Respiradouros Hidrotermais	17
3.3	<i>Download</i> de dados animais marinhos	17
3.4	Mapeamento de habitat	18
3.5	Sistema de acostagem	19
3.6	Micro Rov Simulator	20
3.7	ROV Design And Analysis Toolbox	20
3.8	ROVsim	21
3.9	Exemplo de sistema LBL	23
4.1	Arquitectura de um INS	27
4.2	Referenciais do corpo e de navegação	28
4.3	<i>North East Down</i>	29
4.4	Graus de liberdade do veículo	30
5.1	Referencial Global e referencial local do robô diferencial	40
6.1	Arquitectura de Software	43
6.2	Arquitectura de Hardware	44
6.3	Tampa do sistema auxiliar de instrumentação	46
6.4	Cilindro do sistema auxiliar de instrumentação	46
6.5	IN: Manoeuvre Based Mission Control System for Autonomous Surface Vehicle	48
6.6	Exemplo seguimento de trajectória	49
6.7	Exemplo seguimento recta	50
6.8	Exemplo seguimento recta com $e_\psi = 0$	50
6.9	Calota esférica	52
6.10	Sistema USBL <i>Micron Nav</i>	53
6.11	Arquitectura do sistema USBL	53
6.12	Exemplo Autómato Híbrido	55
6.13	Exemplo de implementação de Autómato Híbrido	55
6.14	Exemplo de um Bloco <i>StateFlow</i>	57
6.15	Implementação exemplo <i>StateFlow</i>	58
6.16	Resultado da implementação exemplo <i>StateFlow</i>	58

6.17	Resumo do algoritmo de <i>Kalman</i>	63
7.1	Leituras USBL em x	68
7.2	Leituras USBL em y	69
7.3	Diagrama do sistema implementado	70
7.4	Sistema Auxiliar de Instrumentação	70
7.5	Software Micro	72
7.6	Simulador	73
7.7	Trajectória dos veículos	73
7.8	Bloco equações movimento ROV	74
7.9	Bloco ASV	75
7.10	Diagrama de estados do bloco controlador	75
7.11	Trajectória do ASV e ROV	76
7.12	Erro e sinais de <i>enable</i>	77
7.13	Limite de operação <i>vs</i> profundidade	78
7.14	Trajectória do ASV e ROV (modo autónomo)	78
7.15	Erro e sinais de <i>enable</i>	79
7.16	Limite de operação <i>vs</i> profundidade	79
7.17	Arquitectura Filtro <i>Kalman</i>	81
7.18	Resultado Filtro <i>Kalman</i>	82
7.19	Resultado Filtro <i>Kalman</i> em y	83
7.20	Resultado Filtro <i>Kalman</i> em x	83
7.21	INS sem filtro de <i>Kalman</i>	84
7.22	Inovação	84
7.23	Resultado Filtro <i>Kalman</i> (recta)	84
7.24	Integração em termos de <i>hardware</i>	85
7.25	Integração em termos de <i>software</i>	85
7.26	Posição do ASV e ROV no mundo	87
7.27	Posição do ASV e ROV no mundo(<i>google maps</i>)	87
7.28	ROAZ em missão	88
7.29	Lançamento do ROV através do ASV	88
7.30	Fotograma do vídeo da inspecção	89

Lista de Tabelas

4.1	Diferentes níveis de INS	28
4.2	Notação usada	29
6.1	Características do sistema USBL	53
7.1	Estrutura da mensagem do sistema USBL	68
7.2	Massa adicionada	80
7.3	Coefficientes de <i>drag</i> lineares	80
7.4	Coefficientes de <i>drag</i> quadráticos	80

Esta página foi intencionalmente deixada em branco.

Acrónimos

2D	Bidimensional
3D	Tridimensional
AOSN	Autonomous Ocean Sampling Network
ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
CAN	Controller Area Network
CTD	Conductivity, Temperature, and Depth
DC	Direct Current
DOF	Degrees Of Freedom
DVL	Doppler Velocity Log
FEUP	Faculdade de Engenharia da Universidade do Porto
GPS	Global Positioning System
I ² C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISEP	Instituto Superior de Engenharia do Porto
LBL	Long BaseLine
LiFePO ₄	Lithium Iron Phosphate
LSA	Laboratório de Sistemas Autónomos
NED	North East Down
RDA	ROV Design And Analysis Toolbox
ROV	Remotelly Operated Vehicle
SCOUT	Surface Craft for Oceanographic and Undersea Testing
SLAM	Simultaneous Localization And Mapping
SPI	Serial Peripheral Interface Bus
UAV	Unmaned Aerial Vehicle
USB	Universal Serial Bus
USBL	Ultra-Short BaseLine
VBAP	Virtual Body Artificial Potencial

Esta página foi intencionalmente deixada em branco.

Capítulo 1

Introdução

1.1 Âmbito da Dissertação

Os veículos robóticos marinhos, assumem um papel de crescente importância nas diferentes aplicações relacionadas com o mar. Os robôs de uma forma geral e os robôs marinhos em particular, oferecem um vasto conjunto de soluções para a resolução de muitos dos problemas que emergem no mundo actual. Os ASV, veículos autónomos de superfície (de origem inglesa *Autonomous Surface Vehicle*) e os ROV, veículos operados remotamente (*Remotely Operated Vehicle* da literatura anglo saxónica) são usados em áreas tão distintas como a aquacultura, a monitorização ambiental, inspecção de infra-estruturas subaquáticas, busca e salvamento, oceanografia ou batimetria. Neste contexto a investigação, nos tópicos associados como o controlo e a navegação, assume uma importância elevada dado o número e relevância de aplicações possíveis.

A utilização de um ROV a partir de um veículo de superfície quer de forma semi-autónoma, tele-operada ou autónoma, permite estender a gama de aplicações do conjunto, bem como a execução de um novo conjunto de missões. Desde a inspecção de estruturas com o veículo de superfície a fornecer suporte logístico até à operação autónoma de um robô de superfície em missões com recolha de informação sensorial subaquática. A operação de um ROV está sempre ligada à necessidade de operação através de utilizador humano, e existem situações, em que não é possível fazer chegar a embarcação de suporte à zona de operação.

Um cenário em que a coordenação entre um ASV e um ROV é vantajosa é a inspecção de um exutor submarino. Um exutor submarino pode estender-se-se por diversos

quilómetros tornando a sua inspecção uma manobra complexa e com uma logística elevada. A coordenação de um ASV com um ROV é conveniente uma vez que se pode pilotar o ROV sem nunca ocorrerem problemas como a falta de cabo ou a necessidade de mudar a estação base, uma vez que o ASV pode autonomamente seguir o ROV servindo assim de plataforma móvel de apoio. Esta solução permite adicionalmente operar o ROV de forma convencional em tele-operação mas fazê-lo a partir de uma localização remota, tirando partido das funcionalidades autónomas do robô de superfície. Também é possível a realização de missões de forma autónoma, utilizando *feedback* sensorial (subaquático) para o controlo do sistema, uma vez conhecida *à priori* a localização da estrutura a inspecionar.

O presente trabalho endereça a problemática da integração de um sistema ROV com o veículo de superfície ROAZ. Em particular vamos dedicar alguma atenção à definição de um conjunto de manobras coordenadas, bem como ao sistema de navegação do ROV.

1.2 Objectivos e Contribuições

O objectivo desta dissertação visa abordar o problema de coordenação entre um ASV e um ROV, permitindo a realização de missões operacionais com os dois veículos integrados bem como a implementação futura de modos de controlo autónomo avançado. O presente trabalho pretende contribuir nas seguintes linhas e etapas:

- Obter um modelo do veículo subaquático em seis graus de liberdade por forma a poder desenvolver e validar soluções de controlo coordenado;
- Desenvolver uma arquitectura de controlo coordenado permitindo a realização de manobras coordenadas do sistema;
- Implementar um simulador de forma a validar as manobras coordenadas;
- Implementar um sistema de navegação relativa do veículo ROV face ao de superfície recorrendo a um sensor acústico USBL e um sensor INS no ROV;
- Instrumentar o ROV com INS para o sistema de navegação relativa do veículo submarino;
- Integrar o ROV com ASV nas componentes de Hardware e Software.

1.3 Organização da Tese

Esta dissertação está organizada da seguinte forma:

O primeiro capítulo apresenta o problema endereçado bem como a sua motivação e objectivos do trabalho.

No segundo capítulo é discutido o problema de coordenação de um veículo operado remotamente com um veículo autónomo de superfície.

O terceiro capítulo apresenta alguns exemplos de coordenação multi robótica mais relevantes no estado da arte bem como alguns ambientes de simulação relacionados.

Os métodos e técnicas de navegação robótica utilizadas no presente trabalho são abordados no capítulo quatro. A modelização do veículo subaquático (ROV) e do de superfície é abordada no capítulo seguinte.

A arquitectura de controlo proposta, exemplos de manobras de controlo coordenadas bem como o sistema de navegação relativa projectados para solução do problema são apresentados no capítulo 6.

O sétimo capítulo é dedicado à apresentação dos resultados obtidos na implementação do projecto desde a validação das manobras de controlo em ambiente de simulação desenvolvido para o efeito aos testes realizados em ambiente real.

O ultimo capítulo apresenta a conclusões e as perspectivas de trabalho futuro.

Esta página foi intencionalmente deixada em branco.

Capítulo 2

Problema de Coordenação entre ASV e ROV

2.1 Formulação do Problema

Existem hoje soluções de veículos autónomos que passam por ASV, AUV (Autonomous Underwater Vehicle) e ROV para efectuar diversos trabalhos aquáticos. Destes são exemplos batimetria e operações de salvamento [1, 2] entre outros.

Na actualidade a maior parte das inspecções são feitas por mergulhadores. Mas este tipo de inspecção acarreta grandes desvantagens: áreas de inspecção reduzida, impossibilidade de operação em locais adversos, inspecção efectuada por peritos e a informação (ex. fotos e vídeos) não está disponível em tempo real bem como estão bastante condicionadas com as condições da água como a temperatura e a visibilidade. Este tipo de inspecções com ser humanos também têm limitações ao nível da quantidade de sensores que o mergulhador pode levar e a qualidade e quantidade de dados recolhidos.

O uso de veículos robóticos de superfície origina algumas vantagens como a possibilidade de obter mapas do fundo marinho completamente georreferenciados, com recurso a vários sensores como são exemplos sonares de varrimento lateral (*sidescan*) e multi-feixe (*Multibeam*). O facto do veículo poder ser autónomo é por si só uma grande vantagem, o veículo autónomo tem a capacidade de recolher informação, processar essa mesma informação, interpretar os dados tudo isto com a participação humana mas não intervenção humana. No entanto o robôs de superfície têm a desvantagem de ter uma perspectiva limitada, ou seja, todos os sensores montados a bordo de um ASV dão uma perspectiva

de cima do fundo escondendo por vezes elementos tais como grutas que estão na zona de oclusão dos sensores.

A inspecção ou recolha de dados com AUV, também é usada hoje em dia[3]. Os AUV são capazes de cobrir grandes áreas e permitem incorporar múltiplos sensores, mas não são capazes de fornecer dados em tempo real e possuem limitação de energia e autonomia.

Tendo em conta tudo o que foi descrito em cima o conceito de ter um ASV e um ROV a trabalhar em conjunto faz todo o sentido uma vez que os dois juntos minimizam as desvantagens de existir um só veículo para inspecção. O esquema ASV/ROV traz desde logo a vantagem, em relação ao ASV, o facto do veículo de superfície passar a ser dotado de visão subaquática e móvel. O conjunto é capaz de inspecionar estruturas periodicamente com reduzida intervenção humana, sendo também capaz de produzir dados em tempo real. O ASV pode ser usado como *relay* de comunicações de forma a ser possível trabalhar com o ROV em zonas de difícil acesso, o ROV pode ser equipado com um grande conjunto de sensores para trabalharem em conjunto com os sensores do ASV em tarefas tais como: mapeamento 3D de estruturas, exploração e georreferenciação de lugares de interesse, inspecção a grandes profundidades, missões de batimetria, inspecção vídeo no espectro visível, infravermelhos e acústico.

No caso do veículo de superfície servir de *relay* de comunicações para o ROV é um exemplo dos diversos cenários de aplicação coordenada ASV/ROV. Nesta situação o ROV é operado remotamente, o sistema de apoio ao ROV encontra-se a bordo do ASV e na estação base existe o interface com o ROV que permite trocar informação com o ROV. Esta situação tem especial interesse se estivermos perante uma situação de catástrofe em que seja exigido algum tipo de trabalho ou inspecção num local demasiado perigoso.

Um outro exemplo de grande importância do ponto de vista da exploração da vida marinha dos oceanos é por exemplo o mapeamento de habitat. Neste cenário a operação pode ser totalmente autónoma. O ROV efectua um trajecto pré-definido e o ASV acompanha o ROV por forma a este ter sempre cabo disponível para a sua operação ou o ASV pode manter-se sempre por cima do ROV.

Considerando um cenário operacional como representado na figura 2.1 com um veículo autónomo de superfície e um ROV a realizar uma manobra coordenada (como por exemplo a inspecção de uma estrutura subaquática), impõe uma série de requisitos.

Por um lado, sob o ponto de vista de controlo é necessário gerar os sinais de controlo

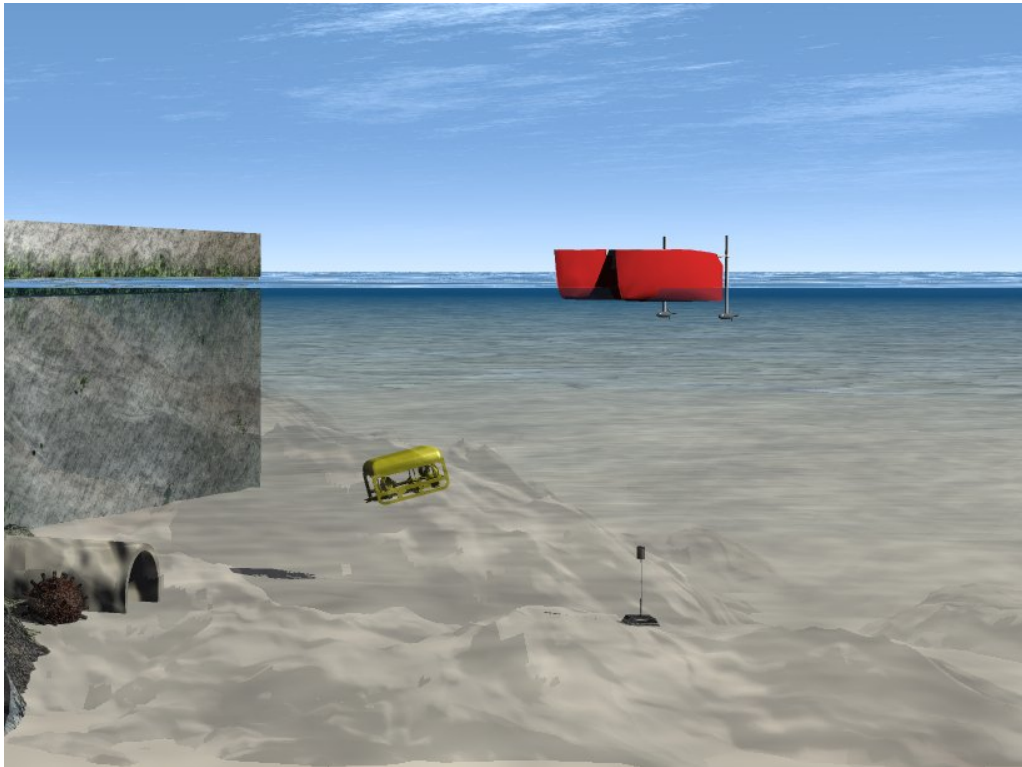


Figura 2.1: cenário operacional

quer para o veículo de superfície quer para o ROV de acordo com os objectivos da missão. Por outro é necessário a existência de um sistema de navegação integrado que permita posicionar o ROV relativamente ao ASV e dependendo da missão georreferenciar informação sensorial do ROV. Temos pois 2 problemas:

- Controlo - determinar os sinais de controlo dos dois veículos de forma coordenada, para atingir um objectivo global do conjunto;
- Navegação - determinar o estado/posição dos dois veículos, em particular a posição e atitude do veículo subaquático em relação ao veículo de superfície bem como a georreferenciação de *features*.

Em termos de controlo, o problema é pois: dados objectivos específicos de missão, aqui representados por $Man_{sistema}$ (ver secção 6.1), obter os controlos do veículo (ver capítulo 5), neste caso:

$$u_{asv} = [u_1, u_2]^T \text{ (sinais de controlo para os motores do ASV)}$$

$$u_{ROV} = \tau = [X, Y, Z, N] \text{ (no caso do ROV o controlo lateral } Y \text{ nem sempre é possível)}$$

Para o subproblema de localização a operação conjunta dos dois robôs requer a possibilidade de estimar o estado dos veículos, neste caso:

$$X_{ASV} = [X, Y, Z, \theta, \varphi, \psi, U, V, W, p, q, r] \text{ e}$$

$$X_{ROV} = [X, Y, Z, \theta, \varphi, \psi, U, V, W, p, q, r]$$

Neste trabalho são endereçados os dois problemas. É proposta uma arquitectura de controlo bem como exemplos de manobras coordenadas e também apresentar um solução de navegação relativa envolvendo um sistema acústico USBL e sensores inerciais.

Podem-se definir um conjunto de manobras possíveis para o sistema ASV/ROV nomeadamente:

- Teleoperação - Nesta manobra o ROV é controlado directamente pelo operador, servindo o ASV de *relay* de comunicações
- Teleoperação Assistida - Neste manobra o ASV segue o ROV de forma a este nunca ter problemas de falta de cabo
- Operação Autónoma - Neste modo os dois veículos funcionam de forma autónoma.

A localização do ROV em relação ao ASV é de importância vital para que o sistema funcione. O ASV tem de servir de suporte ao ROV em tudo o que este necessita, comunicações, energia e sistema de recolha.

2.2 Cenários Operacionais De Coordenação ASV-ROV

A coordenação entre um ASV e um ROV tem diversas aplicações em cenários reais. Nas secções abaixo são enumeradas as aplicações mais relevantes de um sistema ASV-ROV.

2.2.1 Inspeção Teleoperada De Estruturas

Na inspeção de estruturas os ROV [4, 5] são hoje uma ferramenta usada com muita frequência. São alguns exemplos a inspeção de paredes nos portos, cascos de navios, molhes, cais entre outros. A coordenação de um ASV com um ROV é de especial importância nestes cenários uma vez que permite, desde logo, que o ASV seja a base logística/operacional. A inspeção remota é também uma das vantagens bem como o acesso imediato aos responsáveis e a georeferenciação das estruturas inspeccionadas.

2.2.2 Inspeção De Estruturas Extensas

A utilização de um sistema ASV-ROV pode revelar-se aqui bastante vantajoso. Neste tipo de inspeções o sistema ASV-ROV pode ser totalmente autónomo, mantendo ainda assim a capacidade de o utilizador intervir quando necessário. Neste tipo de inspeção, como exemplo temos as paredes de um porto. É atribuída uma missão ao ROV e ASV e os dois em conjunto realizam a missão e podem enviar os dados em tempo real ao mesmo tempo que guardam os dados para pós-processamento. Note-se que a operação em modo autónomo ou semi-autónomo permite aumentar a eficiência face à tele-operação com o ROV com base em terra ou com base num veículo tripulado.

2.2.3 Arqueologia Marinha

Os ROV desempenham um papel importante no que diz respeito à arqueologia marinha. Recentemente temos como exemplo uma expedição aos destroços do *Titanic* em que foi utilizado um ROV, o remora 6000. Este ROV estava equipado com câmaras de alta definição, equipamento acústico 3D e um sonar de varrimento lateral.

2.3 ROAZ

O veículo ROAZ II é um veículo autónomo de superfície desenvolvido no LSA/ISEP. O ASV ROAZ II está equipado com diversos sensores entre os quais: *GPS*, *INS*, *Multibeam sonar*, *SideScan Sonar*, Altímetro, Câmara e Radar.

O sistema GPS é constituído por uma antena L1/L2 e um módulo receptor capaz oferecer posição GPS a uma taxa até 10Hz, da empresa *Septentrio*. O sistema INS é um sistema da empresa *Microstrain* e o modelo 3DM-GX1, este sistema está equipado com três acelerómetros, três giroscópios e três magnetómetros. Relativamente ao *Multibeam* e *Sidescan Sonar* e altímetro são todos fabricados pela *imagenex* e são respectivamente os modelos 837B Delta T e *Sportscan*. O Radar é o modelo *RD218 Radome Radar Scanner* da empresa *raymarine* tem uma potência de 2KW. As baterias são baterias *LiFePO₄* de 12V 100Ah. Os motores são dois motores Torquedo de 6 hp. O computador de bordo consiste numa única *motherboard* de baixo consumo e é responsável pelo controlo da missão e navegação de ASV. Este computador trabalha sob o um sistema operativo *Linux* modificado.

O veículo ROAZ II tem como base mecânica um casco do tipo catamaran e foi assim concebido de forma a ser mais estável em missões tanto em cenários de rio ou de mar. O veículo tem um peso de aproximadamente 200 kg e as suas dimensões são: 4,5 metros de comprimento, 2 metros de largura e aproximadamente 1,5 metros de altura.

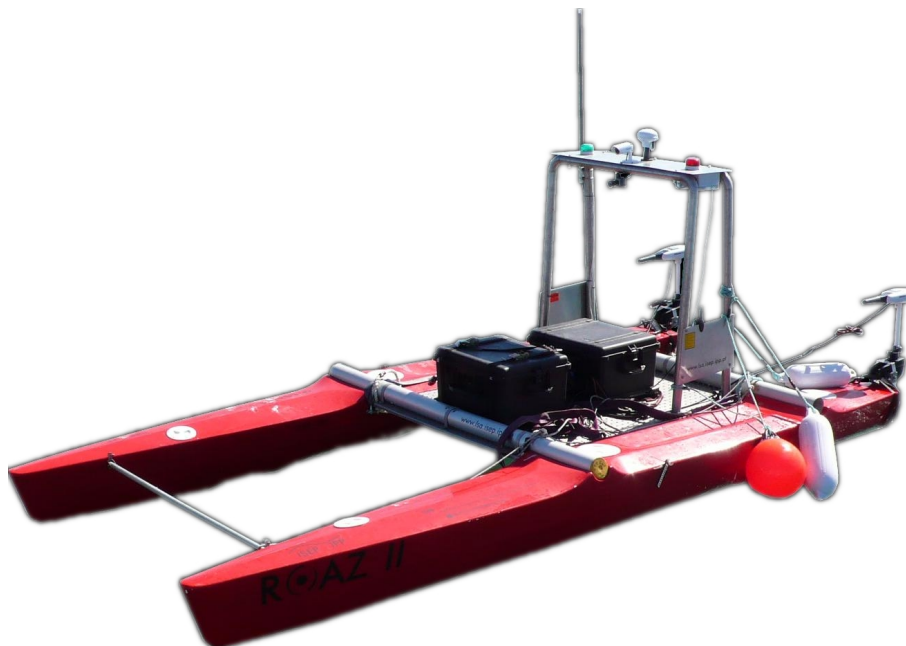


Figura 2.2: Veículo ROAZ II

2.4 Sistemas Robóticos Submarinos do LSA-ISEP

Os dois ROV existentes no LSA-ISEP são o *VideoRay PRO III* (figura 2.3) e o *SeaBotix* (figura 2.4) LBV150.

O ROV *VideoRay* possui 3 motores, 1 a estibordo, outro a bombordo e 1 vertical que permite um controlo em *Surge*, *Heave* e *Yaw* (ver capítulo 5), por sua vez o veículo *SeaBotix* possui 4 motores, 1 a estibordo, 1 a bombordo, 1 vertical e 1 lateral que permite controlo em *Surge*, *Heave*, *Sway* e *Yaw*.

O ROV *Seabotix* ainda não está totalmente dominado em termos de protocolo de comunicação, ao invés do ROV *VideoRay* em que o protocolo de comunicação com os sensores e actuadores a bordo é conhecido. Assim o sistema de controlo coordenado a desenvolver tem de: gerir os actuadores dos ROV de acordo com o objectivo da missão bem como posicionar os ROV relativamente ao ASV.



Figura 2.3: *VideoRay* PRO III



Figura 2.4: *SeaBotix* LBV150

Esta página foi intencionalmente deixada em branco.

Capítulo 3

Estado da Arte

3.1 *Autonomous Surface Vehicle*

Os veículos marinhos de superfície em particular os ASV são actualmente veículos de grande relevância no panorama da robótica mundial [6]. São usados em funções tão variadas como inspecção, vigilância, oceanografia [1], detecção de minas, são também usados como plataforma de apoio a outros veículos autónomos como por exemplo os AUV e veículos aéreos não tripulados UAV (Unmanned Aerial Vehicle).

Dos imensos ASV existentes actualmente podemos dividi-los em três grandes grupos: os ASV militares [7], comerciais [8] e os de desenvolvimento científico [9]. Estes ASV existem de diversas formas sendo a mais popular o catamaran devido à sua estabilidade, no entanto podemos encontrar ASV na forma de barcos semi-rígidos, muito utilizados nas aplicações militares, sendo também possível encontrar ASV com outros formatos como kayakes [10] e lanchas.

Em termos de fonte de energia a mais usada é a eléctrica (baterias) existindo também exemplos de veículos movidos a diesel ou vento.

Quanto aos veículos militares estes são usados por exércitos como o dos Estados Unidos da América, Israel, Britânico e Paquistânês, entre outros. São usados em missões de reconhecimento, vigilância, segurança de portos ou detecção de minas entre outras. Os veículos *OWL MK II*, *SeaFox MKII*, *USSV-HTF* [11] e o veículo *Protector* são exemplos de veículos com aplicações militares.

Existem inúmeras empresas a produzir ASV comerciais entre elas podemos destacar a empresa Sea Robotics [8] que dedica a sua actividade ao desenvolvimento e construção de

diversos ASV para missões de batimetria e monitorizações ambientais. Do ponto de vista dos ASV de desenvolvimento científico temos o exemplo do ASV ROAZ e ROAZ II do Laboratório de Sistemas Autónomos (LSA) do Instituto Superior de Engenharia do Porto (ISEP). Estes dois veículos permitem validar algoritmos de controlo, coordenação multi-robótica entre outras tecnologias relacionadas com veículos marítimos. Outros exemplos de veículos são Delfim [12], SCOUT (Surface Craft for Oceanographic and Undersea Testing) [10], *AutoCat* [13] e *Aces* [14].

3.2 *Remotely Operated Vehicle*

Os veículos operados remotamente têm a particularidade de estarem ligados à sua base através de um cabo umbilical, o que permite ao ROV ter vantagens como vídeo de alta definição, dados em tempo real, disponibilidade de energia elevada e duração de operação entre outras.

Os ROV à semelhança do que acontece com os ASV podem ter aplicações militares e civis. Nas aplicações militares destacam-se missões como detecção e detonação de minas subaquáticas, como é exemplo o ROV AN/SLQ-48, usado pelo exército dos Estados Unidos da América, para detonação de minas. Em termos civis os ROV são usados maioritariamente em inspeções e intervenções, são exemplos de ROV de inspeção VideoRay PRO III [15], Seabotix LBV150 [16] ou Phantom [17] no que diz respeito a ROV de intervenção temos como exemplos os ROV Hercules [18], Triton [19] e Bathysaurus XL [20]. Nestas missões os ROV são capazes de ser equipados com diversos sensores e ferramentas entre eles sonares, braços de corte, braços de trabalho, magnetómetros ou CTD.

3.3 **Múltiplos Veículos Robóticos Em Ambiente Marinho**

A coordenação de múltiplos veículos autónomos em ambiente marinho permite a realização de missões complexas com inerentes vantagens face à utilização de sistema com um só veículo. No caso de um veículo falhar temos os outros veículos para desempenhar a tarefa que estava atribuída ao veículo que falhou. Algumas desvantagens de usar um único veículo fortemente equipado ao invés de usar múltiplos veículos são desde logo a falta de robustez face às possíveis falhas no sistema e problemas de autonomia energética que ocorrem para missões com grande extensão ou duração.

Pioneiro na utilização de múltiplos sistemas robóticos no mar o projecto *AOSN* (*Autonomous Ocean Sampling Network*) [21] é um projecto que tem como objectivo recolher dados do oceano de forma autónoma. Na sequência com o *AOSN II* [22] a coordenação de múltiplos veículos subaquáticos é efectuada com um método para controlo cooperativo de múltiplos robôs denominado de *VBAP* (*Virtual Bodies and Artificial potenciales*). [22]

Este método utiliza o conceito de corpo artificial (*Virtual Body*) que consiste em pontos de referência, ligados entre si, que se movem e são denominados de *Virtual Leaders*. De forma aos veículos seguirem uma trajectória coordenada cada veículo apenas tem de seguir *Virtual Leaders* não necessitando assim de saber a posição dos restantes veículos na formação. Para que a formação funcione perfeitamente é necessário manter a distância dos veículos entre si e entre os veículos e os líderes artificiais para isso é usado o algoritmo de potenciais artificiais. [23] Na Figura 3.1 está representado o método VBAP em 2D, nesta

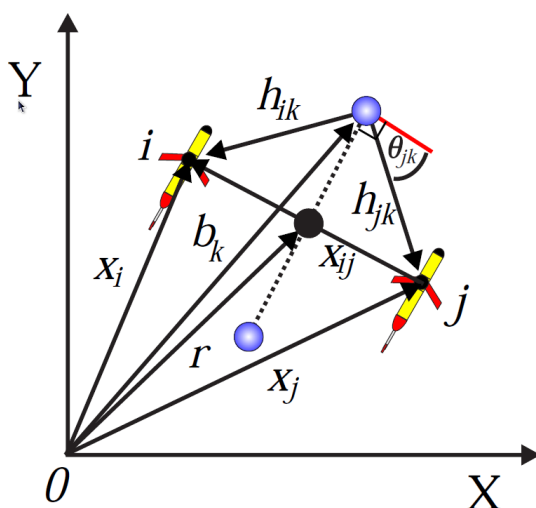


Figura 3.1: Representação 2D de VBAP

figura os círculos sombreados representam os *Virtual Leaders*. Assim temos que X_i e X_j são os vectores de posição dos veículos desde a origem do referencial inercial por sua vez o vector B_k representa o vector posição do *Virtual Leader*. $X_{ij} = X_i - X_j$ e $h_{ik} = x_i - b_k$ são respectivamente a distância entre os veículos e a distância do veículo i ao *Virtual Leader*.

No contexto Europeu existem vários projectos de investigação quer a nível internacional quer a nível nacional que visam o desenvolvimento de sistemas multiroboticos em ambiente marinho. Um exemplo é o GREX que tem como objectivo o desenvolvimento de um quadro conceptual e de *middleware* de forma a coordenar múltiplos robôs heterogéneos

que trabalham de forma a atingir um objectivo comum de forma otimizada.

Os robôs marinhos constituem uma fonte típica de estudo e desenvolvimento pelos problemas associados que não se encontram a uma taxa muito mais elevada que em robôs terrestres que são, por exemplo, falha de comunicação, navegação por GPS inexistente (veículos submarinos), perturbações externas imprevisíveis. A resolução destes problemas pode ser vantajoso para outras áreas que não só a robótica marítima. [24]

A coordenação de múltiplos veículos traz uma série de desafios para a comunidade científica. Mas ainda antes da coordenação de vários veículos existe o problema de controlar o movimento de um só veículo que podemos dividir em três grandes grupos: i) Estabilizar o veículo num determinado ponto com uma determinada atitude ii); seguimento de trajectória com restrições temporais; iii) seguimento de trajectória do ponto de vista geométrico sem restrições temporais. [25]

No panorama actual os problemas de controlo de um único veículo estão evoluídos de forma a ser possível partir para os problemas de controlo de múltiplos robôs. Exemplos (alguns dos quais tópicos de estudo de projectos em curso) com cenários de aplicação de múltiplos veículos marinhos são abordados em seguida.

3.3.1 Respiradouros Hidrotermais

Neste primeiro cenário os veículos adaptam a sua formação de forma a seguirem o gradiente de concentração de metano com o objectivo de encontrar a sua origem tal como apresentado na figura 3.2. Neste primeiro cenário os veículos executam esta tarefa levando em conta as leituras dos seus sensores bem como as leituras dos sensores dos outros veículos.

3.3.2 Descarga De Dados De Peixes

Neste caso ASV e AUV trabalham em conjunto de forma a captar informação de etiquetas de animais marinhos. Vários ASV tentam percorrer uma zona de forma a encontrar sinais acústicos de etiquetas de animais, uma vez encontrado o ASV em cooperação com outros ASV encontram a posição do animal e enviam um sinal de comando para que um AUV siga esse animal de forma a fazer *download* da informação contida na etiqueta. A ilustração deste cenário está representada na figura 3.3.

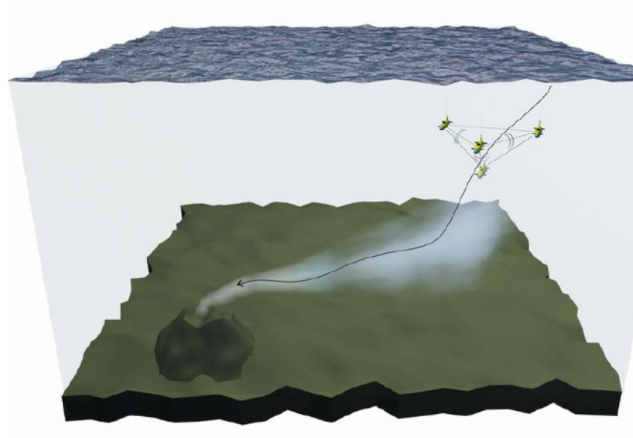


Figura 3.2: Busca de Respiradouros Hidrotermais

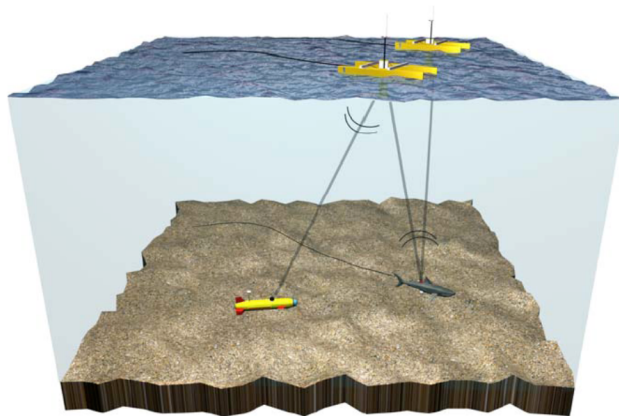


Figura 3.3: *Download* de dados animais marinhos

3.3.3 Mapeamento De *Habitat*

Um ROV ligado a um ASV inspecciona o fundo do oceano usando o ASV como ponte de comunicações. Uma vez encontrado algo de relevante na zona são enviados AUV para essa zona de forma a recolher dados sobre o local tal como apresentado na figura 3.4. Este cenário revela-se de particular interesse no que diz respeito a este trabalho, uma vez que o movimento coordenado entre o ASV com o ROV é especialmente necessário.

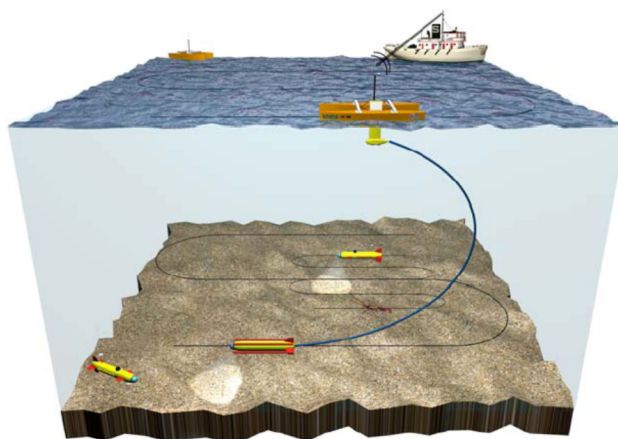


Figura 3.4: Mapeamento de habitat

3.4 Coordenação ASV AUV

Inserido no projecto ROAZ do LSA do ISEP está o sistema de acostagem de um AUV usando a informação visual [26]. Este trabalho consiste num sistema de baixo custo que através da informação do sistema de visão é capaz de guiar o ASV até este estar perfeitamente alinhado com o AUV e efectuar a acostagem. Os testes deste sistema revelaram que o ASV é capaz, em cenário operacional, de fazer a acostagem do AUV apenas usando a visão para esse fim, através de um filtro de Kalman é possível estimar a posição do AUV e com isso o ASV faz as manobra de aproximação e acostagem do AUV. A Figura 3.5 mostra este sistema num cenário operacional de rio.

Na FEUP (Faculdade de Engenharia da Universidade do Porto) foi desenvolvido um sistema em que um ASV tem como função seguir um AUV e manter-se sempre directamente por cima do AUV, isto é vantajoso uma vez que as maiores taxas de comunicação acústica são conseguidas quando os transdutores estão alinhados na vertical. [27] Neste



Figura 3.5: Sistema de acostagem

trabalho são usados dois veículos: o ASV ZARCO [28] e o AUV MARES [29]. O AUV para se localizar interroga 4 bóias que estão localizadas na área de operação, todas as bóias recebem a interrogação mas apenas a bóia interrogada responde o que faz com que o AUV tenha total controlo das comunicações acústicas, por sua vez o ASV escuta os sinais acústicos trocados entre o AUV e as diferentes bóias e estima a posição do AUV sendo para isso necessário que o AUV sempre que recebe uma mensagem acústica de uma bóia demora um período de tempo determinístico a enviar nova interrogação.

3.5 Simuladores

3.5.0.1 *Micro Rov Simulator*

O simulador *Micro ROV Simulator* é um pequeno simulador desenvolvido no âmbito de uma dissertação de mestrado da universidade de Zagreb [32]. Este simulador apenas tem a capacidade para simular o ROV *VideoRay Pro II*. Não implementa a dinâmica do ROV apenas implementa a parte da cinemática e virtualiza os instrumentos do ROV. Uma funcionalidade interessante deste simulador é usar as medidas dos sensores e mostrar, através do seu módulo de realidade virtual, uma percepção de como está o ROV no mundo (posição e atitude) dando deste modo ao piloto uma perspectiva diferente da realidade. Neste momento o simulador ainda se encontra em fase de desenvolvimento mas já tem implementadas as seguintes características: gestão da câmara, intensidade das luzes, gestão da garra, sentido de rotação dos trusters. A imagem 3.6(a) mostra o sistema implementado

desenhados de raiz testar algoritmos de controlo. De seguida são apresentados alguns dos simuladores comerciais existentes na actualidade.

3.5.2.1 ROVsim

O simulador ROVsim, da empresa *Marine Simulation LLC* [31], apresenta-se como sendo um simulador de treino e não um simulador de carácter científico, mesmo assim a física presente no simulador é bastante precisa ao nível dos melhores simuladores 3D existentes. Este simulador tem 3 versões diferentes em termos de produtos para o consumidor, existem as versões: *ROVsim O&G*, *ROVsim Pro* e *Undersea Pilot Series*. Na versão *ROVsim O&G* foi especialmente desenhada para dar resposta às necessidades da indústria do petróleo, já a *ROVsim Pro* vai de encontro às necessidades dos profissionais da área de robótica submarina e por fim a versão *Undersea Pilot Series* é a versão mais económica com o básico para o treino de pilotagem em ROV.

Este simulador tem como principais objectivos e funcionalidades o treino de pilotos, plataforma de testes e simular sensores. A figura 3.8 demonstra algumas das funcionalidades que a interface do simulador possui.



Figura 3.8: ROVsim

3.6 Navegação Subaquática

Existem hoje uma variedade de técnicas de Navegação subaquática [33]. O método mais simples de obtenção de localização consiste em integrar a velocidade de forma a obter estimativas da posição. A medição das diferentes componentes da velocidade é conseguida usando uma bússola e um sensor de velocidade da água. Mas existe o problema das correntes marítimas que vão ser acrescentadas às diferentes componentes da velocidade e adicionam erro às medidas.

Com sistemas INS (Inercial Navigation System) as acelerações nos diferentes eixos dos veículos são integradas duas vezes por forma a obter as componentes da posição do veículo. O erro nestes sistemas pode atingir vários quilómetros numa hora o que faz destes sistemas pouco fiáveis em termos de navegação para missões de longa duração.

Outra tecnologia usada é usando sensores Doppler Velocity Log (DVL) [34] que permitem ao veículo calcular a sua velocidade relativamente ao fundo do oceano, este sistema tem boa precisão quando usado em conjunto com o sistema INS mas possui a desvantagem de só poder ser usado em missões perto do fundo do oceano.

Sistemas como o GPS são bons para obter a posição do veículo de forma precisa. Para veículos de superfície como os ASV são uma boa solução de localização no entanto apresentam enormes inconvenientes no que diz respeito a veículos submarinos. Resulta assim a necessidade do veículo emergir várias vezes durante a missão de forma a obter a sua posição com qualidade, para que isto aconteça o veículo gasta uma quantidade considerável de energia só para emergir e submergir. Por outro lado a natureza da missão pode não permitir ao veículo esta actividade.

A energia electromagnética não consegue viajar grandes distâncias na água excepto se for a frequência muito baixas, por sua vez a energia acústica é capaz de viajar com facilidade, então faróis acústicos podem ser uma boa solução de navegação subaquática sem necessidade do veículo ter forçosamente de vir à superfície. Nas técnicas de localização acústica as duas mais populares são Long BaseLine(LBL) [35] e Ultra Short BaseLine(USBL) [36, 37]. O método LBL utiliza transdutores externos que são instalados em localizações bem conhecidas. O veículo interroga os diferentes transdutores e através do tempo de resposta para cada um deles calcula o ponto de intercepção de várias esferas obtendo assim a posição. Por sua vez os sistemas USBL utilizam um conjunto de hidrofones instalados no veículo que recebem informação acústica de um farol acústico

externo. Através da diferença de fase e tempo de voo dos sinais recebidos em cada um dos hidrofones é possível determinar a distância e ângulo de veículo face ao farol.

Outros métodos usados para navegação subaquática passam por métodos geofísicos, isto é, se o veículo possuir um mapa *à priori* do local onde vai efectuar a missão é possível fundir a informação que obtém dos seus sensores com o mapa que possui e desta forma calcular a sua posição. No entanto um mapa de qualidade pode ser difícil de obter.

Mais recentemente começam a ser usados métodos como a hodometria visual [38]. A hodometria visual é o processo que através da análise de imagens sucessivas é capaz de determinar a posição e orientação da câmara.

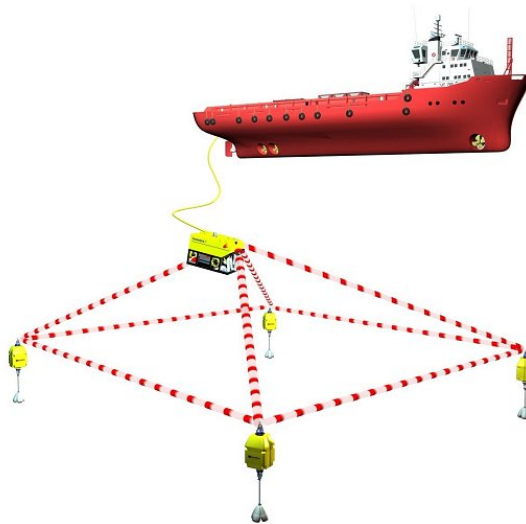


Figura 3.9: Exemplo de sistema LBL

Esta página foi intencionalmente deixada em branco.

Capítulo 4

Fundamentos de Navegação

O termo navegação pode ser definido como sendo o processo de planejar, identificar e controlar o movimento do veículo, aéreo ou marítimo, desde o seu ponto de partida até ao seu destino. Este capítulo aborda alguns conceitos básicos de navegação indispensáveis à realização deste trabalho.

4.1 Técnicas De Navegação

4.1.1 Sistemas Baseados Em Pontos De Referência

Os sistemas de navegação baseados em pontos de referência são os mais utilizados para a navegação de veículos autónomos. Estes sistemas disponibilizam informação relativa à posição e orientação com uma elevada precisão e processamento mínimo. Sensores como ultra-sons, rádio frequência, fontes luminosas entre outros são utilizados por estes sistemas de forma a determinar a posição e orientação do veículo relativamente a um ponto de referência. A posição do ponto de referência pode ser conhecida no mundo, tratando-se assim de um “active beacon”, neste método o cálculo da posição é feito utilizando conceitos de triangulação e trilateração. No caso, da posição não ser conhecida no mundo, mas sim de forma relativa estamos perante um sistema de navegação “*dead reckoning*”, em que a informação é referenciada em relação a um referencial local. Em ambos os casos apresentados anteriormente, estão muitas vezes associados à criação de mapas para navegação. Estes mapas utilizam a informação de posição e orientação do veículo, e dos pontos de referência para representar o mundo e a atitude do veículo. A técnica de SLAM (*Simultaneous Localization and Mapping*) [39] permite a integração de todas as

características de navegação, ou seja, com a técnica de SLAM é possível localizar o veículo e ao mesmo tempo gerar um mapa do mundo, permitindo ao robô navegar nesse mapa.

4.1.2 Sistemas "Dead Reckoning"

"Dead Reckoning" é uma técnica de cálculo de posição a partir de medições da velocidade. Como consequência disto a navegação é efectuada na ausência de posições fixas e consiste em calcular a posição através da integração da velocidade (estimada ou medida directamente), assim a posição obtida é em relação à posição inicial que é conhecida. Este tipo de navegação é feito com base na informação de sensores como por exemplo: *encoders*, tacómetros, bússola ou IMU ("Inertial Measurement Unit").

Estes sistemas têm como principal vantagem o facto dos sensores utilizados estarem contidos no próprio robô e não necessitarem de nenhum apoio exterior ou estruturação do meio, no entanto, uma grande desvantagem é o facto das medidas estarem contaminadas de ruído, e este seja acumulado, originando erros acentuados por integração das medidas.

4.1.3 Sistemas de Navegação Inercial (INS)

O sistema de Navegação Inercial (INS) é por si só um sistema de navegação autónomo, que fornece informação dinâmica sobre a posição e velocidade do veículo, tendo como base as medidas obtidas de um IMU("Inertial Measurement Unit"). Um IMU é constituído por uma tríade de acelerómetros e giroscópios colocados de forma ortogonal, em alguns casos o IMU possui um magnetómetro por forma a calcular a atitude de forma mais precisa, a partir do campo magnético da terra.

O principio básico de operação da navegação inercial é baseado na lei de movimento de *Newton*, que diz que um objecto se mantém num estado de repouso ou movimento uniforme, até ser perturbado por uma força exterior. A aplicação de uma força exterior gera a aceleração, que é medida pelos acelerómetros presentes no IMU. Esta aceleração quando integrada duas vezes dá a informação da alteração da posição do veículo em relação às condições iniciais(4.1). Deste modo, o INS fornece informação relativa à alteração do estado do veículo, pelo que as condições iniciais (posição, velocidade e atitude) do veículo têm de estar bem definidas para uma correcta navegação.

O processo de cálculo dos valores de posição, velocidade e atitude, nem sempre é muito simples devido ao facto de que o IMU fornece medidas relativas a um dado referencial que

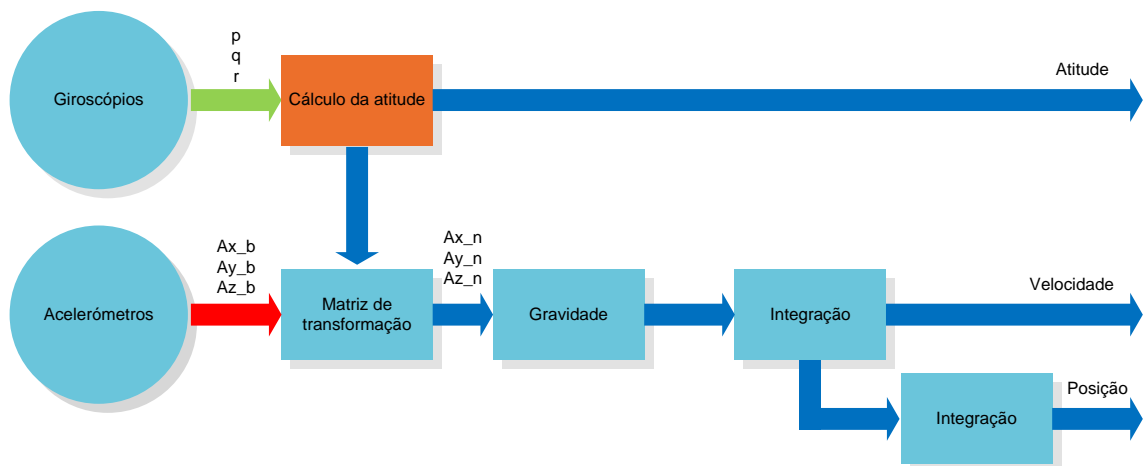


Figura 4.1: Arquitectura de um INS

na maioria das vezes é diferente do referencial usado na navegação. Por esta razão o INS contém giroscópios, que servem para fornecer medições das velocidades angulares, para que seja possível transformar a aceleração medida no referencial de navegação antes do processo de integração. Assim, a precisão dos giroscópios é um factor determinante para a qualidade geral de um INS, porque erros relacionados com a transformação da aceleração entre os referenciais usados provocam um erro na velocidade e posição que é acumulado ao longo do tempo.

O desempenho de um sistema inercial, tanto dos giroscópios como dos acelerómetros, é afectado por uma variedade de erros [40]. A maioria dos erros pode ser classificada no *bias*, factor de escala, desalinhamento de eixos e ruído. O *bias* é um erro presente quer nos giroscópios quer nos acelerómetros e consiste em duas partes: uma determinística denominada de *bias offset* e uma parte aleatória denominada de *bias drift*. O *bias offset* refere-se ao *offset* presente nas medidas dos sistema inercial, este *offset* pode ser determinado através de uma calibração do sistema inercial. Por sua vez o *bias drift* refere-se á taxa com que o erro se vai acumulando com o tempo.

De acordo com [41] existem quatro níveis de IMU (a classificação baseia-se na precisão do sistema): nível estratégico, nível de navegação, nível táctico e nível de consumidor (tabela 4.1).

Nível de IMU	Estratégico	Navegação	Tático	Consumidor
Erro de Posição (km/h)	<0.03	<4	18.5 até 40	>40
Erro giroscópio (°/h)	0.0001	0.015	1 até 10	1000
Erro acelerometro (mg)	0.001	0.1	1.0	20

Tabela 4.1: Diferentes níveis de INS

4.2 Sistemas de Coordenadas

4.2.1 Referenciais

Na representação das grandezas físicas envolvidas no modelo do veículo são utilizados 2 sistemas de coordenadas: um fixo ao veículo x_b, y_b, z_b e um sistema de coordenadas considerado inercial fixo à terra, o referencial NED. O referencial fixo ao veículo, que se pode observar na figura 4.2, possui o eixo x_b alinhado com direcção longitudinal do veículo, o eixo y_b perpendicular a x_b o eixo z_b aponta para baixo.

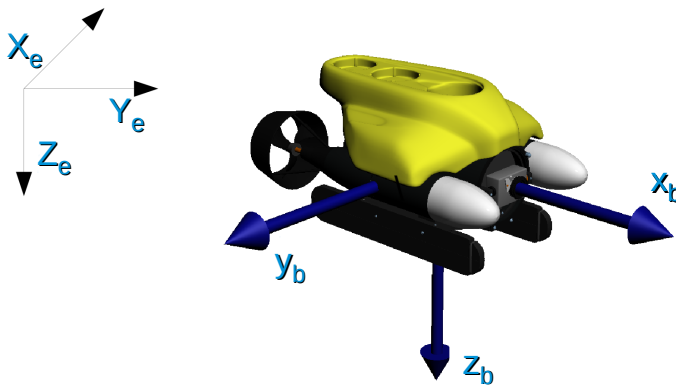
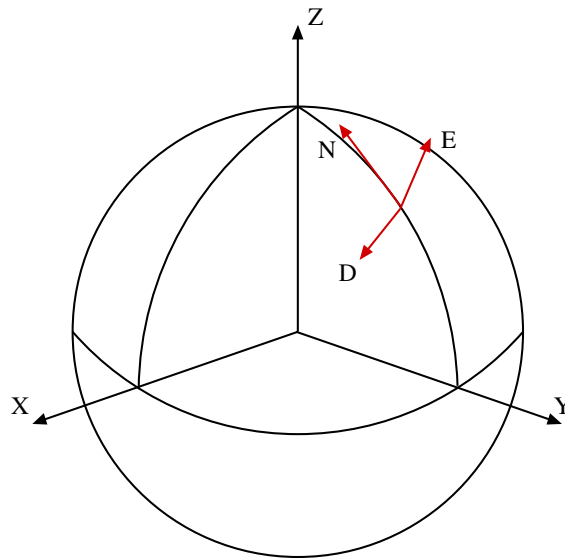


Figura 4.2: Referenciais do corpo e de navegação

O referencial NED (*North - East - Down*) é um referencial muito usado em navegação. Este referencial é um referencial ortogonal, constituído por 3 vectores: N(*north*) que aponta para o pólo magnético norte da terra, E (*east*) que aponta para o Este geodésico da terra e por fim o vector D que aponta no sentido da gravidade local (figura 4.3).

Figura 4.3: *North East Down*

4.3 Notação Usada

Na tabela 4.2 está descrita a notação usada neste capítulo que foi proposta por [42] e usado em [43], na figura 4.4 estão representados graficamente os diferentes graus de liberdade do veículo. Na modelação de veículos subaquáticos é conveniente trabalhar com dois referenciais, um fixo ao veículo (x_b, y_b, z_b) e outro inercial (x, y, z) , assim a dinâmica do veículo é modelada no referencial do veículo sendo simplificada pelo facto que os momentos de inércia do veículo permanecem constantes em relação ao referencial do veículo seja qual for a sua orientação.

Na tabela seguinte (4.2) podemos observar a notação usada na representação das diversas grandezas utilizadas na modelação do veículo. As velocidades estão representadas no referencial do veículo e as posições estão representadas no referencial inercial. Em termos

	Forças e Momentos	Velocidade Linear e angular	Posição e ângulos de Euler
Movimentos segundo a direcção x (<i>surge</i>)	X	u	x
Movimentos segundo a direcção y (<i>sway</i>)	Y	v	y
Movimentos segundo a direcção z (<i>heave</i>)	Z	w	z
Rotação em torno do eixo x (<i>roll</i>)	K	p	ϕ
Rotação em torno do eixo y (<i>pitch</i>)	M	q	θ
Rotação em torno do eixo z (<i>yaw</i>)	N	r	ψ

Tabela 4.2: Notação usada

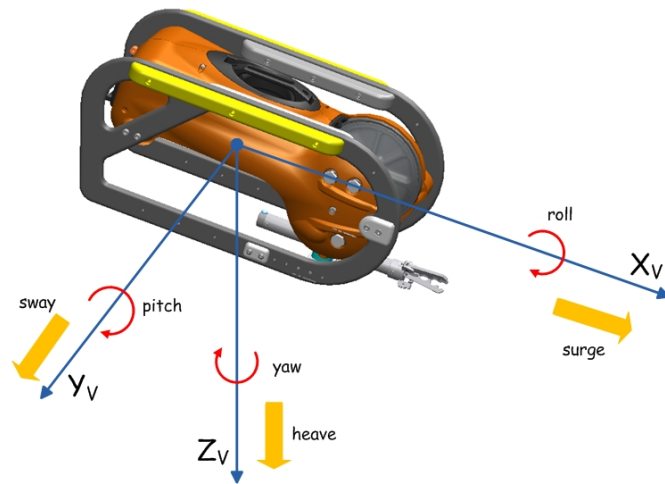


Figura 4.4: Graus de liberdade do veículo

vectoriais as grandezas acima descritas tomam a seguinte forma

$$\eta_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}; \quad \eta_2 = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}; \quad \eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}; \quad (4.1)$$

$$\nu_1 = \begin{bmatrix} u \\ v \\ w \end{bmatrix}; \quad \nu_2 = \begin{bmatrix} p \\ q \\ r \end{bmatrix}; \quad \nu = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix}; \quad (4.2)$$

$$\tau_1 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}; \quad \tau_2 = \begin{bmatrix} K \\ M \\ N \end{bmatrix}; \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (4.3)$$

em que η_1 e η_2 são a posição e orientação, respectivamente, do referencial do veículo em relação ao referencial inercial; ν_1 e ν_2 são as velocidades linear e angular no referencial fixo ao corpo e por fim τ_1 e τ_2 são as forças e momentos exercidos sobre o veículo no referencial do veículo.

4.3.1 Cinemática

É possível relacionar um vector expresso no referencial do veículo (x_b, y_b, z_b) com o referencial inercial (x, y, z) através de 3 rotações elementares em torno dos eixos (x_b, y_b, z_b) . Estas rotações são efectuadas através dos ângulos de Euler para cada eixo, o eixo x_b é rodado de ϕ , o eixo y_b é rodado de θ e z_b é rodado de ψ as rotações são efectuadas pela ordem que foram apresentadas¹. As matrizes de rotação obtidas segundo cada eixo são

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \quad (4.4)$$

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (4.5)$$

$$R_{z,\psi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

Tal como definido em [43] a matriz de transformação entre os referenciais tem a seguinte forma apresentada em 4.7, para simplificar a leitura $\cos \cdot = c \cdot$ e $\sin \cdot = s \cdot$

$$\begin{aligned} J_1(\phi, \theta, \psi) &= J_1(\eta_2) = \\ &= \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \end{aligned} \quad (4.7)$$

J_1 é ortonormal, isto é, $J_1^{-1} = J_1^T$.

A velocidade linear ν_1 no referencial do corpo relaciona-se com a velocidade $\dot{\eta}_1$ no referencial inercial pela expressão 4.8

$$\dot{\eta}_1 = J_1(\eta_2)\nu_1 \quad (4.8)$$

¹A ordem das rotações é importante uma vez que não são comutativas

De forma análoga também podemos transformar as velocidades angulares através da matriz $J_2(\eta_2)$ [43]

$$\dot{\eta}_2 = J_2(\eta_2)\nu_2 \quad (4.9)$$

$$J_2 = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{s\phi}{c\theta} & \frac{c\phi}{c\theta} \end{bmatrix} \quad (4.10)$$

Em relação a 4.10 podemos afirmar que existe uma singularidade² para $\theta = \pm 90^\circ$ e que J_2 não é ortogonal $J_2^{-1} \neq J_2^T$. A expressão 4.11 relaciona as velocidades lineares e angulares nos dois referenciais.

$$\dot{\eta} = J(\eta_2)\nu \quad J(\eta_2) = \begin{bmatrix} J_1(\eta_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & J_2(\eta_2) \end{bmatrix} \quad (4.11)$$

²Esta singularidade não é relevante uma vez que no caso dos veículos utilizados nesta dissertação nunca apresentam valores de *pitch* próximos de 90°

Capítulo 5

Modelação

Neste capítulo é descrito o processo de modelação de veículos marinhos (ASV e ROV) que conduziram aos modelos de base utilizados quer na simulação quer no projecto de controlo coordenado.

5.1 Modelo Dinâmico

O modelo dinâmico de um veículo marinho é obtido através da modelação de um corpo rígido em 6 graus de liberdade.

5.2 Dinâmica de um corpo rígido

Nesta secção são apresentadas as equações do movimento de translação e rotação de um corpo rígido, estas equações são apresentadas considerando que $\dot{m} = 0$ que o corpo é indeformável e constituído por um número finito de partículas.

Para os 6 DOF de um corpo rígido temos para o movimento de translação a equação 5.1 e para o movimento de rotação a equação 5.2 [43]:

$$m(\dot{v}_0 + w \cdot v_0 + \dot{w} \cdot r_G + w \cdot (w \cdot r_G)) = f_0 \quad (5.1)$$

$$m_0 = I_0 \cdot \dot{w} + w \cdot (I_0 \cdot w) + m \cdot r_G \cdot (\dot{v} + w \cdot \dot{v}_0) \quad (5.2)$$

onde,

I_0 é o tensor de inércia definido na origem do referencial fixo ao corpo;

$r_G = [x_G; y_G; z_G]^T$ é a posição do centro de gravidade do corpo;

$w \equiv \nu_2$ é o vector de velocidade de rotação;

$f_0 =$ é a soma das forças externas que actuam no corpo;

$m_0 =$ é a soma dos momentos externos que actuam no corpo;

$\dot{v}_0 \equiv \dot{v}_1$ é a derivada em ordem ao tempo da velocidade no referencial fixo ao corpo.

Decompondo as forças e os momentos segundo os eixos do referencial do corpo e dos eixos de rotação do corpo obtemos aplicando a notação descrita em 4.3:

$$X = m[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] \quad (5.3)$$

$$Y = m[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] \quad (5.4)$$

$$Z = m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(rq + \dot{p})] \quad (5.5)$$

$$\begin{aligned} K = I_x \dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} + \\ + m[y_G(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)] \end{aligned} \quad (5.6)$$

$$\begin{aligned} M = I_y \dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} + \\ + m[z_G(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] \end{aligned} \quad (5.7)$$

$$\begin{aligned} N = I_z \dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} + \\ + m[x_G(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)] \end{aligned} \quad (5.8)$$

A dedução detalhada destas equações pode ser encontrado em [43]. Escolhendo a origem do veículo com a origem do referencial do corpo, temos $I_{xz} = I_{yz} = I_{xy} = 0$. Podemos agora obter a dinâmica do corpo rígido de forma matricial como

$$M_{RB} \dot{\nu} + C_{RB}(\nu)\nu = \tau_{RB} \quad (5.9)$$

onde,

$$M_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & -mz_G & my_G & I_{xx} & 0 & 0 \\ mz_G & 0 & -mx_G & 0 & I_{yy} & 0 \\ -my_G & mx_G & 0 & 0 & 0 & I_{zz} \end{bmatrix} \quad (5.10)$$

M_{RB} é conhecida como a matriz de inércia do corpo rígido

$$C_{RB} = \begin{bmatrix} 0 & 0 & 0 & m(y_G q + z_G r) - m(x_G p - w) & -m(x_G r - v) \\ 0 & 0 & 0 & -m(x_G q - w) m(z_G r + x_G p) & -m(y_G r - u) \\ 0 & 0 & 0 & -m(x_G r - v) - m(z_G q + u) m(x_G p + y_G + q) \\ -m(y_G q + z_G r) & m(y_G p + w) & m(z_G p - v) & 0 & I_{zz} q & -I_{yy} p \\ m(x_G q - w) & -m(z_G r + x_G p) & -m(z_G q + u) & -I_{zz} q & 0 & I_{xx} p \\ m(x_G r + v) & m(y_G r - u) & -m(x_G p + y_G q) & I_{yy} q & -I_{xx} p & 0 \end{bmatrix} \quad (5.11)$$

C_{RB} é denominada de matriz de termos de *Coriolis* e centrípetos.

5.3 Forças Hidrodinâmicas

5.3.1 Massa Acrescentada

Quando um corpo se encontra em movimento inserido num fluído estacionário, gera um movimento das partículas que o envolvem. São induzidas forças e momentos contrários ao movimento do corpo devido à aceleração das partículas do fluído. Estas forças e momentos são conhecidos como massa acrescentada.

As forças e momentos que actuam no corpo devido ao efeito da massa acrescentada são dados por:

$$\tau_A((\dot{v}), v) = -M_A \dot{v} - C_A(v)v \quad (5.12)$$

τ_A é o vector de forças e momentos de massa acrescentada total. M_A é dada por:

$$M_A = \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \quad (5.13)$$

A matriz de termos de *Coriolis* e centrípetos, $C_A(v)$ é dada por:

$$C_A(v) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \quad (5.14)$$

em que

$$a_1 = X_{\dot{u}}u + X_{\dot{v}}v + X_{\dot{w}}w + X_{\dot{p}}p + X_{\dot{q}}q + X_{\dot{r}}r;$$

$$a_2 = X_{\dot{v}}u + Y_{\dot{v}}v + Y_{\dot{w}}w + Y_{\dot{p}}p + Y_{\dot{q}}q + Y_{\dot{r}}r;$$

$$a_3 = X_{\dot{w}}u + Y_{\dot{w}}v + Z_{\dot{w}}w + Z_{\dot{p}}p + Z_{\dot{q}}q + Z_{\dot{r}}r;$$

$$b_1 = X_{\dot{p}}u + Y_{\dot{p}}v + Z_{\dot{p}}w + K_{\dot{p}}p + K_{\dot{q}}q + K_{\dot{r}}r.$$

$$b_2 = X_{\dot{q}}u + Y_{\dot{q}}v + Z_{\dot{q}}w + K_{\dot{q}}p + M_{\dot{q}}q + M_{\dot{r}}r;$$

$$b_3 = X_{\dot{r}}u + Y_{\dot{r}}v + Z_{\dot{r}}w + K_{\dot{r}}p + M_{\dot{r}}q + N_{\dot{r}}r;$$

5.3.2 Amortecimento Hidrodinâmico

O amortecimento hidrodinâmico ($D(v)$) para veículos marítimos é causado principalmente pelas seguintes componentes: atrito na camada exterior do corpo, à pressão criada pelo movimento e ao arrasto devido aos vórtices criados pelo movimento relativo do fluido na superfície do corpo.

Uma aproximação que é geralmente usada para veículos subaquáticos, considerando que o veículo se move a baixa velocidade tem 3 planos de simetria e ignorando os termos de ordem superior à segunda ordem, é uma matriz diagonal onde figuram apenas os termos lineares e quadráticos como está representado na equação 5.15.

$$D(v) = D_l - D_q \quad (5.15)$$

$$D_l = \begin{bmatrix} X_u & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r \end{bmatrix}$$

$$D_q = \begin{bmatrix} X_{u|u}|U| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{v|v}|V| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{w|w}|W| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{p|p}|P| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{q|q}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{r|r}|r| \end{bmatrix}$$

A soma de todas as forças e momentos devido ao amortecimento hidrodinâmico é dada pela equação que se segue

$$\tau_D = -D(v)v \quad (5.16)$$

5.3.3 Forças Hidrostática

Nos veículos subaquáticos, as forças gravitacionais e de flutuação são normalmente denominadas de forças hidrostática ou de restabelecimento. As forças gravitacionais actuam sobre o centro de gravidade ($r_{CM} = [x_{CM}, y_{CM}, z_{CM}]^T$) enquanto que as forças de flutuação actuam sobre o centro de flutuação ($r_{CF} = [x_{CF}, y_{CF}, z_{CF}]^T$) e está representado em 5.17.

$$g(\eta) = \begin{bmatrix} (W - B)\sin(\theta) \\ -(W - B)\cos(\theta)\sin(\phi) \\ -(W - B)\cos(\theta)\cos(\phi) \\ y_{CF}B\cos(\theta)\cos(\phi) - z_{CF}B\cos(\theta)\sin(\phi) \\ -z_{CF}B\sin(\theta) - x_{CF}B\cos(\theta)\cos(\phi) \\ x_{CF}B\cos(\theta)\sin(\phi) + y_{CF}B\sin(\theta) \end{bmatrix} \quad (5.17)$$

5.4 Equações Do Movimento

As equações do movimento resultantes atendendo ao que foi explicado nas secções anteriores estão descritas em baixo

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau \quad (5.18)$$

$$\dot{\eta} = J(\eta)v \quad (5.19)$$

onde:

$$M = M_{RB} + M_A$$

$$C(v) = C_{RB}(v) + C_A(v)$$

$$D(v) = D_l(v) + D_q(v)$$

$g(\eta)$ é a matriz de forças de restabelecimento

τ é o vector das forças dos propulsores

$J(\eta)$ é a matriz de rotação que está descrita na secção [4.3.1](#)

5.4.1 Modelo Reduzido

A distribuição do peso e flutuação do *VideoRay Pro III* fazem com que este retorne sempre à posição em que *roll* e *pitch* são 0, assim sendo é assumido para todo o tempo que $\phi = 0$, $\theta = 0$, $p = 0$ e $q = 0$. Os propulsores do veículo apenas o afectam em termos de *surge*, *heave* e *yaw*.

Atendendo ao que foi descrito anteriormente podemos dividir o movimento em dois sub-sistemas desacoplados:

1. x, y, ψ, u, v, r para o plano horizontal
2. z, w para o plano vertical.

Esta decomposição do movimento suporta a ideia que o movimento em termos de *surge* é feito através de dois propulsores balanceados que se encontram nos dois lados do ROV e qualquer movimento em *yaw* é feito usando o modo diferencial dos mesmos.

As equações resultantes para o plano horizontal são:

$$m_{11}\dot{u} = -m_{22}vr + X_u u + X_{u|u}|u| + X \quad (5.20)$$

$$m_{22}\dot{v} = m_{11}ur + Y_v v + X_{v|v}|v| \quad (5.21)$$

$$J\dot{r} = N_r r + N_{r|r}|r| + N \quad (5.22)$$

em que,

m_{11} = ao elemento 1,1 da matriz de inércia M,

m_{22} = ao elemento 2,2 da matriz de inércia M,

J= Momento de inércia em torno de Z, que é o elemento (6,6) da matriz de inércia M,

$X_u, X_{u|u}$ = coeficientes de *drag* linear e quadrático na direcção de *surge*,

$Y_v, Y_{v|v}$ = coeficientes de *drag* linear e quadrático na direcção de *sway*,

X=Força externa que actua na direcção de *surge*,

N=Momento externo que actua em torno do eixo Z.

A equação resultante para o plano vertical é:

$$m_{33}\dot{w} = Z_w w + Z_{w|w}|w| + Z \quad (5.23)$$

em que, m_{33} = ao elemento 3,3 da matriz de inércia M,

$X_u, X_{u|u}$ = coeficientes de *drag* linear e quadrático na direcção de *heave*,

Z=Força externa que actua na direcção de *heave*.

5.5 Modelo De Um Robô Diferencial

Por motivos de simplificação o ASV vai ser considerado um veículo de tracção diferencial, assim sendo irá ser descrito de seguida a modelação de um veículo diferencial em termos cinemáticos.

Cada motor contribui individualmente para o movimento do veículo e ao mesmo tempo impõe restrições de movimento.

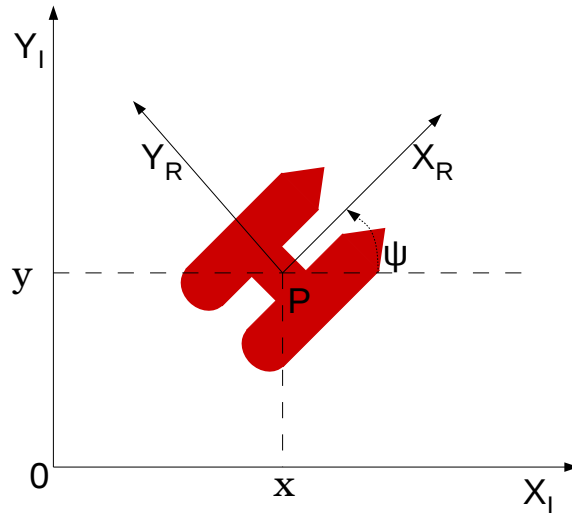


Figura 5.1: Referencial Global e referencial local do robô diferencial

Para modelar o movimento do robô as restrições impostas pelos motores devem ser expressas num referencial apropriado. Assim para especificar a posição do robô vamos considerá-lo como um corpo rígido que se move no plano horizontal e considerá-lo como 3 DOF, (dois para a posição no plano horizontal e 1 para a orientação em torno do eixo vertical). A posição do robô pode ser especificada pela relação entre o referencial global e o referencial local do robô. Na figura 5.1 os eixos X_I e Y_I definem o referencial global enquanto que os eixos X_R e Y_R definem o referencial local. A posição do robô é definida pelas coordenadas x e y do ponto P em X_I, Y_I e pelo ângulo ψ entre os eixos X_I e X_R .

$$P_I = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} \quad (5.24)$$

Num caso geral, o movimento do robô é descrito pelas equações:

$$\dot{x} = u \cdot \cos(\psi) - v \cdot \sin(\psi) \quad (5.25)$$

$$\dot{y} = u \cdot \sin(\psi) + v \cdot \cos(\psi) \quad (5.26)$$

$$\dot{\psi} = w \quad (5.27)$$

na forma matricial:

$$\dot{P}_I = R(\psi) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (5.28)$$

onde u e v são componentes da velocidade linear num referencial instantaneamente sobreposto ao referencial local, w a velocidade angular e R a matriz de rotação ortogonal definida como:

$$R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.29)$$

para um robô diferencial considera-se que :

$$u = (v_r + v_l)/2 \quad (5.30)$$

$$w = (v_r - v_l)/l \quad (5.31)$$

onde v_r e v_l são as velocidades do motor direito e esquerdo e l a distancia entre os motores.

Assim o movimento geral do robô é descrito como dois movimentos instantâneos e elementares: uma translação na direcção de X_I e uma rotação em torno do eixo vertical através do ponto P, de realçar que a velocidade v (velocidade lateral do veículo) é 0 o que resulta nas seguintes equações do movimento:

$$\dot{x} = u \cdot \cos(\psi) \quad (5.32)$$

$$\dot{y} = u \cdot \sin(\psi) \quad (5.33)$$

$$\dot{\theta} = w \quad (5.34)$$

Esta página foi intencionalmente deixada em branco.

Capítulo 6

Projecto

6.1 Arquitectura Do Sistema

A arquitectura do sistema está dividida em duas partes principais: software e hardware.

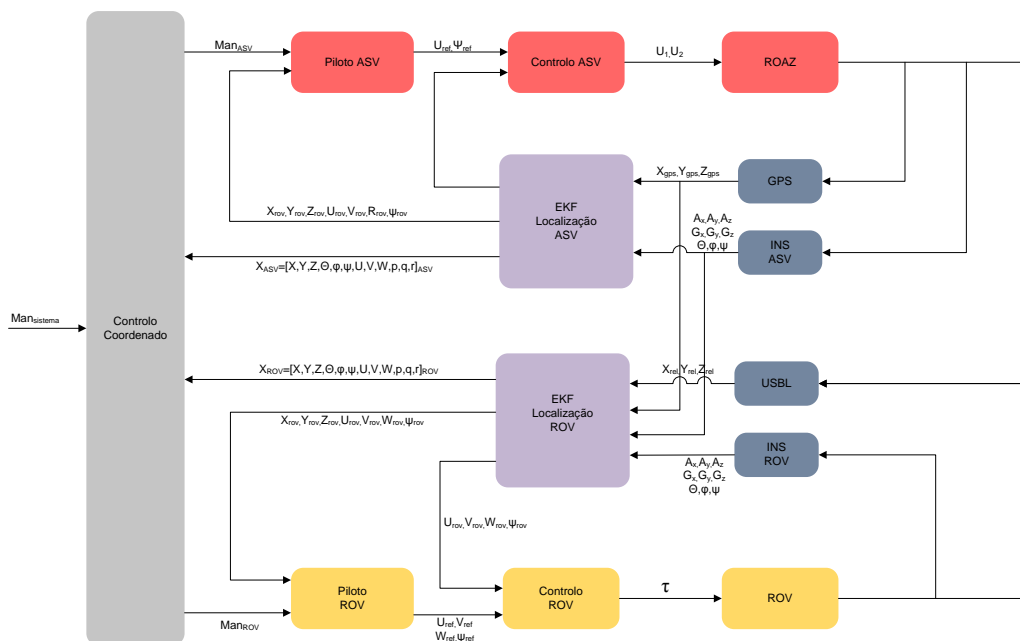


Figura 6.1: Arquitectura de Software

Em termos de software o bloco “Controle Coordenado” funciona como supervisor, recebe uma estrutura “ $Man_{sistema}$ ” que contém os parâmetros da manobra a ser executada pelo sistema e gera 2 estruturas com as manobras para o ASV e para o ROV. Os Blocos “Piloto ROV” e “Piloto ASV” recebem as manobras a executar e geram as referências de

velocidade e orientação para o bloco de controlo, este controla os diferentes motores com base nas referências recebidas. Como podemos observar na figura 6.1 existem dois filtros de localização de forma a localizar os dois veículos. O filtro de localização do ASV recebe a informação proveniente dos sensores INS e GPS e estima a posição e atitude do veículo de superfície. De forma análoga o filtro de localização do veículo subaquático recebe informação dos sensores USBL e INS e estima a posição e atitude do ROV. A informação estimada é usada pelos blocos de Controlo Coordenado, piloto ASV e piloto ROV. O controlo ROV gera os sinais de controlo recebendo como entrada as velocidades e orientação de referência calculadas pelo piloto de acordo com a manobra em execução.

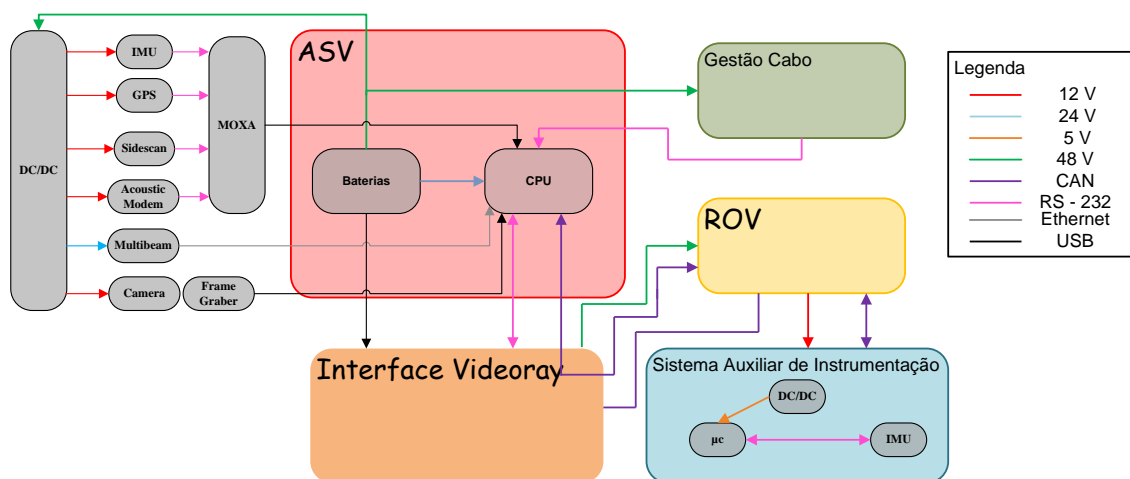


Figura 6.2: Arquitectura de Hardware

Do ponto de vista de hardware existem 3 elementos principais: o ASV, o ROV e o interface ROV. O ASV é composto por um conjunto de sensores, actuadores, baterias e sistema computacional. Sob o ponto de vista de implementação existe um subsistema englobando o sistema computacional de bordo e interfaces e outro com o fornecimento de energia, ambos dispostos em compartimentos estanques separados. Os diversos sensores como podemos observar na figura 6.2 comunicam, alguns deles, com um interface intermédio entre os sensores e o ASV, este interface é um conversor USB-Série e é usado devido à limitação de portas série disponíveis na *board* utilizada. A câmara está ligada a um *frame grabber* que por sua vez liga ao CPU através de um cabo USB. O sensor *Multi-Beam* dispõe de uma ligação *Ethernet* para comunicar com o CPU. Os restantes sensores estão ligados através de uma porta série ao interface *MOXA* que está ligado por USB ao computador.

Em termos de energia o ASV possui 8 baterias (12 V) de LiFePO_4 com uma capacidade de 100 Ah em que a tensão base de funcionamento do ASV são 48 V. Para a alimentação dos diferentes sensores que o ASV possui, existem vários módulos DC/DC para gerar as diferentes tensões necessárias.

O bloco de interface *VideoRay* é o bloco de hardware onde estão os componentes necessários ao funcionamento do ROV, estes componentes foram projectados para um funcionamento em terra e como tal são adaptados para funcionar a bordo do ASV. Este bloco dispõe de DC/DC para alimentar o ROV, um microcontrolador que tem como função fazer a conversão entre os protocolos CAN e RS-232 e um amplificador de vídeo. Adicionalmente existe um sistema de gestão de cabo constituído por um guincho motorizado, sensores de contagem de cabo despendido e controlado pelo sistema computacional do ASV.

6.2 Sistema Auxiliar De Instrumentação

O Sistema Auxiliar de Instrumentação consiste numa compartimento estanque com a finalidade de albergar sensores bem como um sistema computacional para gerir esses sensores. Este sistema é necessário um vez que existem fortes limitações de espaço no ROV. A integração no corpo do veículo dos sensores necessários à navegação relativa, nomeadamente INS e *Pinger* USBL.

6.2.1 Sistema Computacional

O sistema computacional está localizado no Sistema Auxiliar de Instrumentação e consiste num módulo existente no LSA que contém uma unidade de processamento um microcontrolador *Advanced RISC Machine (ARM ® Cortex M3)*, membro da família dos controladores de 32 bits. Este microcontrolador permite comunicar através de diversos protocolos como CAN, USB, RS-232, Ethernet, SPI e I²C.

6.2.2 Controlo De Energia

O sistema de energia existente no sistema auxiliar de instrumentação recebe a energia proveniente do ROV com um tensão de 48 V e converte essa mesma tensão para tensões de 12 V e 5 V de forma a alimentar o microcontrolador e o INS bem como sensores a integrar no futuro.

6.2.3 Hardware

Para a implementação do sistema auxiliar de instrumentação é necessário criar um espaço estanque capaz de levar todos os componentes do sistema em segurança quando submersos. As figuras 6.3 e 6.4 apresentam a solução conceptual do espaço estanque necessário.

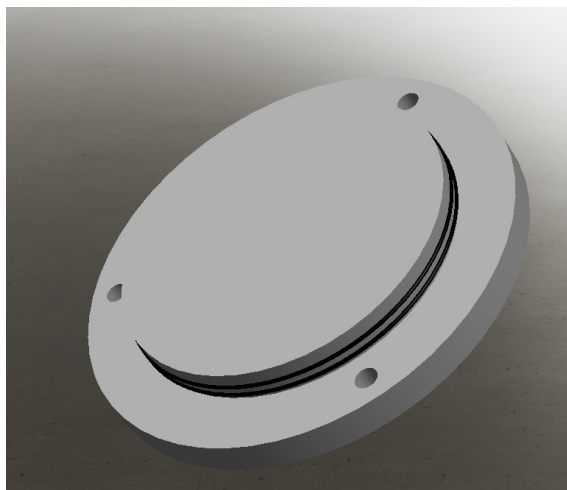


Figura 6.3: Tampa do sistema auxiliar de instrumentação

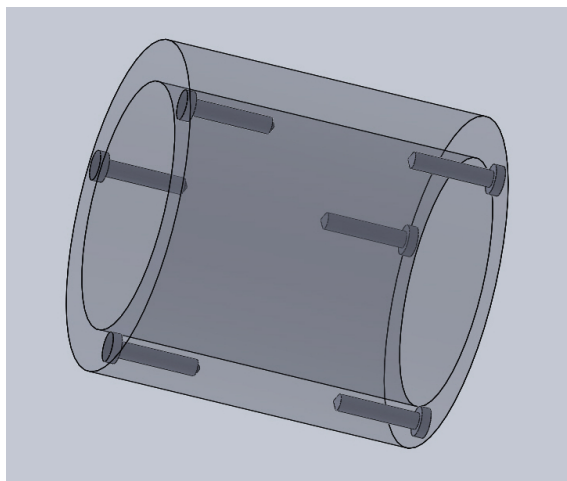


Figura 6.4: Cilindro do sistema auxiliar de instrumentação

Na figura 6.3 é possível observar um *o-ring* duplo de forma a manter a estanquicidade do cilindro. Uma das tampas possui um conector subaquático de forma a fazer o interface entre o hardware que se encontra dentro do cilindro e o ROV. Na figura 6.4 o cilindro está

representado de forma semi-transparente de forma a ser visível o pormenor das furações dos parafusos de fixação das tampas.

6.3 INS

O sistema INS é por si só um sistema de navegação autónomo que fornece informação relativamente à posição, velocidade e atitude de um veículo, esta informação é obtida através de um IMU. Um IMU é um sistema que consiste em três acelerómetros e três giroscópios colocados de forma ortogonal. O princípio de funcionamento baseia-se na primeira lei de Newton, isto é, aplicando uma força a um acelerómetro resulta uma aceleração que pode ser medida e quantificada. Esta aceleração pode ser integrada em ordem ao tempo e obtemos a variação de velocidade que por sua vez integrada em ordem ao tempo fornece uma variação de posição. Para a determinação da velocidade e posição no referencial do veículo de forma absoluta é necessário saber as condições iniciais do veículo.

O sistema INS é um sistema a integrar no veículo ROV de forma a podermos obter a posição, velocidade e orientação do mesmo.

6.4 Arquitectura Do Veículo De Superfície Autónomo ROAZ II

O veículo de superfície ROAZ II possui um sistema computacional de baixo consumo que funciona com um sistema operativo *Linux* modificado. Os blocos de software estão distribuídos tal como estão representados na figura 6.5 [44]. O *software* é composto por módulos que variam entre módulos de baixo nível como interface com os actuadores ou drivers do sistema operativo até módulos de alto nível como a navegação do veículo. O veículo ROAZ II possui a electrónica e as baterias separadas em duas malas distintas. A mala das baterias fornece a energia à mala da electrónica que por sua vez é ela que fornece energia aos sensores e actuadores e faz a interface de comunicação com os sensores. O CPU do ROAZ II tem a capacidade de comunicar com os diversos sensores através de diversos protocolos de comunicação como por exemplo: o *MultiBeam Sonar* comunica através de *Ethernet*, o altímetro através de RS-232 e os controladores dos actuadores através de CAN.

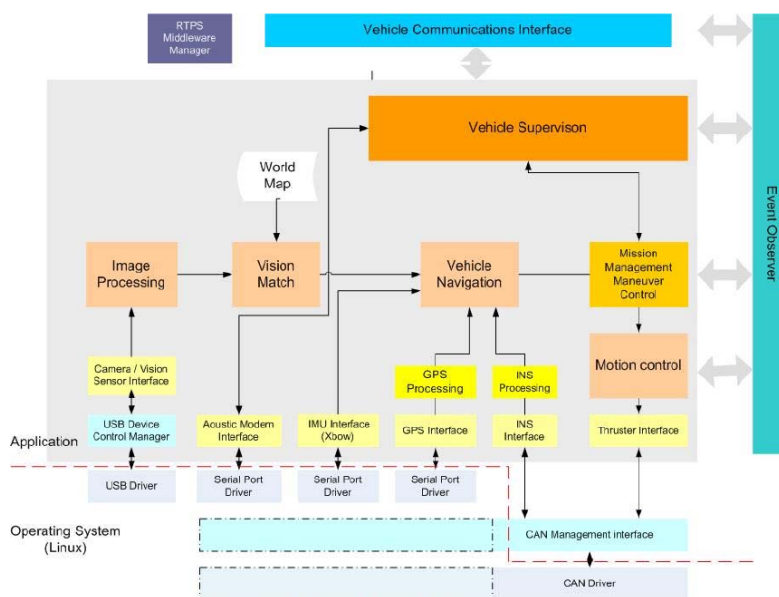


Figura 6.5: IN:Manoeuvre Based Mission Control System for Autonomous Surface Vehicle

6.5 Controlo

6.5.1 Controlo De Trajectória

Num sistema robótico o controlador de trajectória é responsável por gerar referências de posição, velocidade, aceleração e orientação, isto de um ponto de vista genérico. No caso em questão as referências são a posição do veículo, velocidade linear e orientação ψ . Existem inúmeros métodos de controlo de trajectória para veículos de superfície [45, 46, 47, 27]. Iremos usar um esquema de controlo simples baseado na estabilização do erro de orientação e posição em relação ao caminho pretendido.

Considera-se como erro de orientação e_ψ , a diferença entre a tangente à trajectória num ponto e a orientação do veículo. Considera-se também e_d o erro em distância como sendo a menor distância entre o veículo e a trajectória pretendida. Estes dois erros vão influenciar a orientação do robô. Um controlador proporcional pode ser usado com os dois erros acima descritos e com ele os dois ganhos associados:

1. K_d - Convergência para o centro da linha ou trajectória
2. K_ψ - Ângulo ao centro da trajectória

Com uma adequada calibração destes ganhos obtém -se uma trajectória efectuada pelo veículo sem grandes desvios da trajectória pretendida.

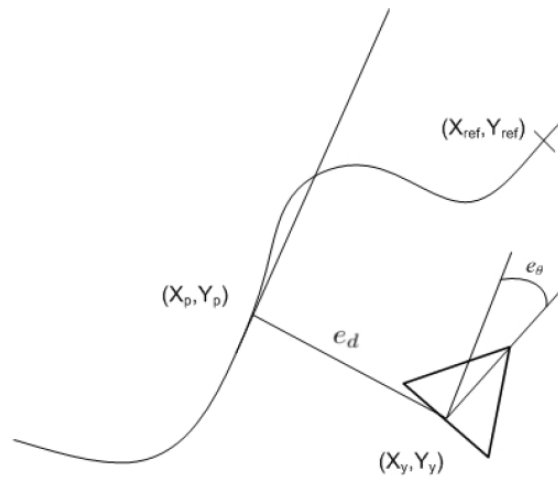


Figura 6.6: Exemplo seguimento de trajetória

6.5.1.1 Orientação em linha recta

No caso particular de uma trajetória recta definida por dois pontos (figura 6.7): ponto inicial (X_i, Y_i) e final (X_f, Y_f) , o valor de referência de orientação é

$$\psi_{ref} = \arctan\left(\frac{Y_f - Y_i}{X_f - X_i}\right) \quad (6.1)$$

assim podemos obter o erro de orientação e_ψ (figura 6.8):

$$e_\psi = \psi_{ref} - \psi_a \quad (6.2)$$

este erro é o valor que o robô tem que variar para que a sua orientação convirja para a orientação de referência (ψ_{ref}) .

A correcção que tem de ser feita em distância é dada pelo erro e_d e é definida como:

$$e_d = -\sin(\psi_{ref}) \cdot (X_a - X_i) + \cos(\psi_{ref}) \cdot (Y_a - Y_i) \quad (6.3)$$

6.5.1.2 Orientação em curva

Na execução de uma trajetória curvilínea, comparativamente à trajetória de uma recta, existe um novo factor a aceleração centrífuga que faz com que o veículo se afaste da trajetória. Uma forma de minimizar esta adversidade é controlar a velocidade dinamicamente, isto é, reduzir a velocidade na entrada da curva e aumentar a velocidade à saída

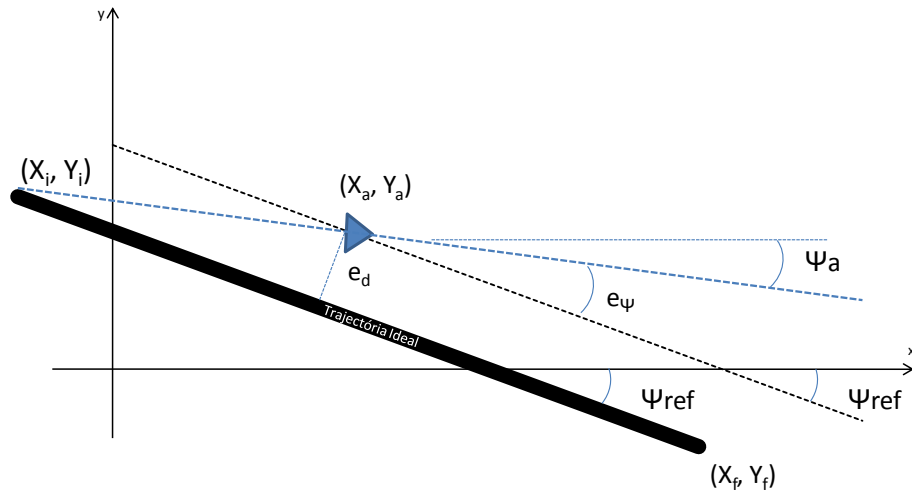


Figura 6.7: Exemplo seguimento recta

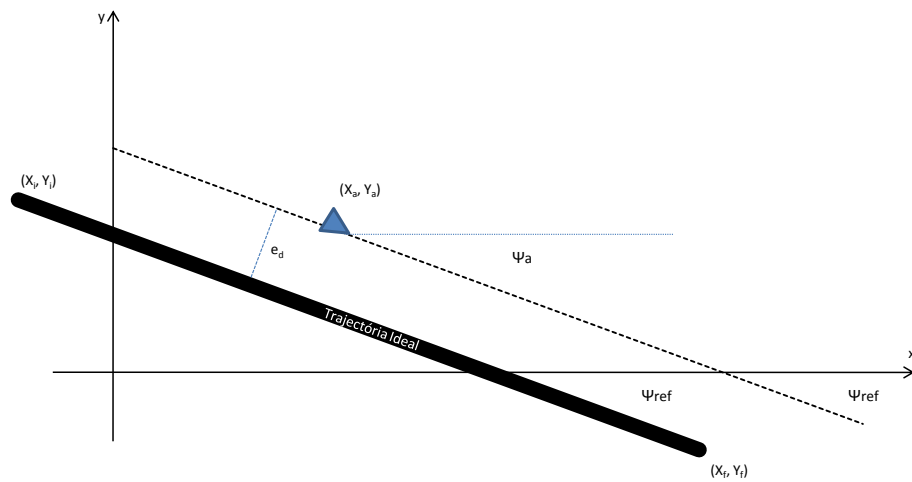


Figura 6.8: Exemplo seguimento recta com $e_\psi = 0$

da curva. O erro e_d é definido como

$$e_d = d \cdot \sqrt{(X_c + X_a)^2 + (Y_c + Y_a)^2} \quad (6.4)$$

onde d é definido como sendo o sentido da curva:

- $d=1$: sentido directo (v e w sinal igual)
- $d=-1$: sentido inverso (v e w de sinal contrário)

a orientação de referência é dada por:

$$\psi_{ref} = \arctan\left(\frac{X_a + X_c}{Y_c - Y_a}\right) \quad (6.5)$$

e o erro de orientação é dado por:

$$e_\psi = \psi_{ref} - \psi_a \quad (6.6)$$

6.6 Coordenação entre ASV e ROV

Uma das desvantagens dos veículos ROV é o facto do seu cabo ser finito, em operações com o ROV os pilotos têm a sua área de actuação limitada pelo cabo do ROV. Para contornar parcialmente esta limitação o ASV desloca-se sempre que o ROV sai da área de actuação. No início de uma missão é definido o raio R de actuação do ROV, assim sendo podemos considerar em 3 dimensões que a área de acção do ROV é uma esfera centrada no ASV e com raio R . Se analisarmos este caso na projecção 2D vendo o ASV de cima a distância do ASV ao ROV depende do raio R , assim o controlador que gere a posição do ASV em relação ao ROV deve ter em conta a profundidade do ROV, ou seja, quanto maior for a profundidade menor será a distância lateral dos dois veículos até ao limite da profundidade ser igual ao R de actuação definido no início da missão. Essa distância pode ser calculada analisando uma calota esférica (figura 6.9)

Usando o teorema de Pitágoras temos que:

$$(R - h)^2 + a^2 = R^2 \quad (6.7)$$

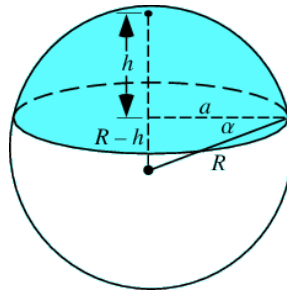


Figura 6.9: Calota esférica

se resolvermos em ordem a a^2 temos:

$$a^2 = 2Rh - h^2 \quad (6.8)$$

a distância a (raio do círculo da base) é:

$$a = \sqrt{h(2R - h)} \quad (6.9)$$

se considerarmos z como $R - h$ no final obtemos:

$$a = \sqrt{R^2 - z^2} \quad (6.10)$$

assim a distância a é a distância máxima, no plano horizontal, que o ROV pode estar do ASV para um raio R e uma profundidade z . Assim o controlador sempre que o ROV chegue ao limite de 95% da distância máxima permitida, pára o ROV e guia o ASV para o ponto onde o ROV se encontra. Com isto numa situação de teleoperação do ROV o piloto apenas tem de se preocupar com a teleoperação do ROV uma vez que o ASV autonomamente irá colocar-se sempre de forma a o ROV puder trabalhar sem a limitação do cabo.

6.7 Navegação USBL

6.7.1 Sistema USBL *Tritech*

O sistema USBL que vai ser usado é o sistema da empresa *Tritech Micron Nav - USBL Tracking System*. Este sistema é constituído por 3 partes: *Micron Nav100 Interface Unit*,

USBL Diving Transducer e *Micron Nav subsea unit* assinalados na figura 6.10 como 1, 2 e 3 respectivamente.



Figura 6.10: Sistema USBL *Micron Nav*

O *Micron Nav100 Interface Unit* e *USBL Diving Transducer* são as partes do sistema que ficam no ASV, o transdutor e a unidade de interface entre o PC e o transdutor, enquanto que o *Micron Nav subsea unit* é a parte do sistema que vai a bordo do ROV, tal como está representado na figura 6.11.

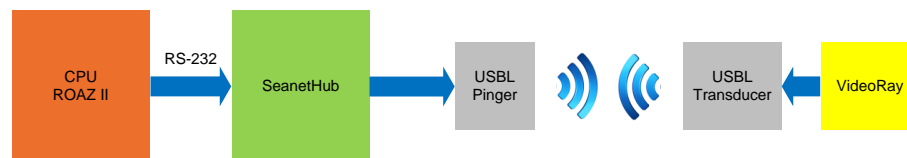


Figura 6.11: Arquitectura do sistema USBL

A tabela 6.1 resume algumas das principais características deste sistema

Intervalo Funcionamento	500 m Horizontal, 150 m Vertical
Precisão Distância	+/- 0,2 metros
Precisão Angular	+/-3 graus
Taxa de Actualização Posição	0,5 Segundos - 10 Segundos
Abertura transdutor	180 graus
Abertura <i>Micron Nav subsea unit</i>	Omnidirecional
Profundidade máxima <i>Micron Nav subsea unit</i>	750 metros

Tabela 6.1: Características do sistema USBL

A unidade *Micron Nav100 Interface Unit* tem a capacidade de integrar outros sensores como GPS e INS, tem também a capacidade de através do protocolo RS-232 disponibilizar mensagens de interesse como por exemplo a posição XYZ do ROV em relação ao ASV.

6.8 Autómato Híbrido

Um sistema híbrido no contexto de sistema de controlo, é um sistema que é composto tanto por componentes contínuos como por componentes discretos. Um exemplo de um sistema híbrido é um motor de um carro cujo bomba injetora (contínuo) é controlada por um microcontrolador (discreto). Uma vez que os sistemas embebidos estão hoje e dia um pouco por toda a parte e cada vez mais assumem um papel importante em sistemas em que a segurança é crítica e a fiabilidade é de extrema importância. Isto requer uma análise rigorosa através de modelos formais, assim para responder a este propósito os autómatos híbridos surgem como a linguagem formal de sistemas híbridos. Como descrito em [48] um autómato híbrido H consiste nos seguintes componentes $H = \{Q, X, Init, In, Dom, R, OUT, E\}$:

- $Q = \{q_1, q_2, \dots\}$ é um conjunto de estados discretos;
- $X = \mathbb{R}^n$ é um conjunto de estados contínuos;
- $Init \subseteq Q \times X$ é um conjunto de estados iniciais;
- In é um conjunto finito de variáveis de entrada;
- $f : Q \times X \times In \rightarrow \mathbb{R}^n$ é um campo de vectores;
- $Dom : Q \rightarrow 2^{X \times In}$ é o domínio;
- $R : Q \times X \times In$ é a relação de *Reset*;
- Out é um conjunto finito de variáveis de saída.

Na figura 6.12 está simulado o funcionamento de um controlador de temperatura, onde x é uma variável de entrada com informação da temperatura ambiente. No modo *off* o controlador está a arrefecer o ambiente de acordo com as condições de fluxo $x' = -0.1x$. No modo *on* o controlador está a aquecer o ambiente de acordo com as condições de fluxo $x' = 5 - 0.1x$. No início a temperatura é de 20 graus, quando a temperatura descer abaixo de 19 graus, conforme a condição de mudança de estado $x < 19$ o controlador passa para o modo *on*. Quando a temperatura subir acima de 21 graus comuta para estado *off*, conforme a condição de mudança de estado $x > 21$.

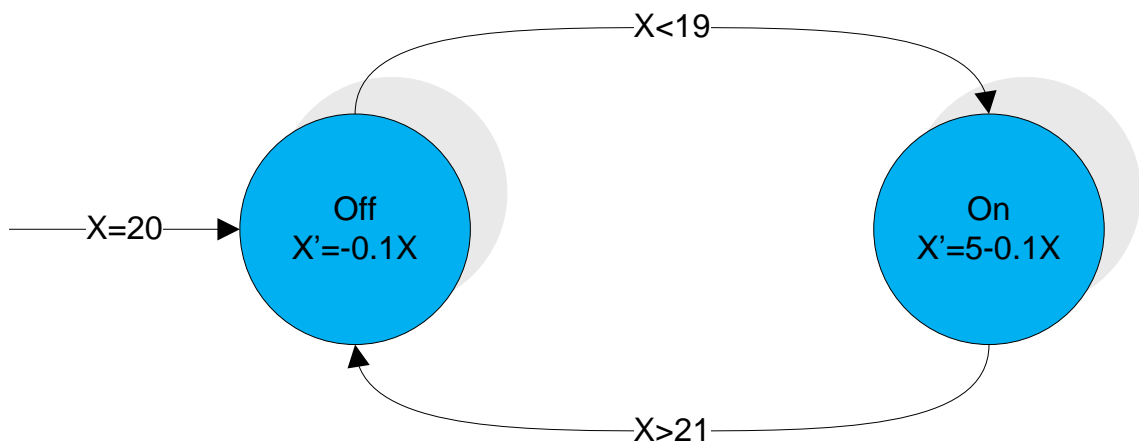


Figura 6.12: Exemplo Autómato Híbrido

6.8.1 StateFlow

Stateflow (segue o formalismo *StateChart* descrito em [49]) é uma ferramenta gráfica que funciona em conjunto com o *simulink* de forma a modelar sistemas orientados a eventos, como são o caso dos autómotos híbridos. Esta ferramenta fornece um editor gráfico, que à semelhança do *simulink*, basta arrastar "Blocos" para a janela principal. O *StateFlow* estende a capacidade de modelação adicionando hierarquia, paralelismo, eventos e acções. Os blocos *StateFlow* são tratados como blocos normais do *simulink*, por esta razão podem ser integrados de forma muito simples com qualquer modelo desenvolvido no *simulink* (Figura 6.13).

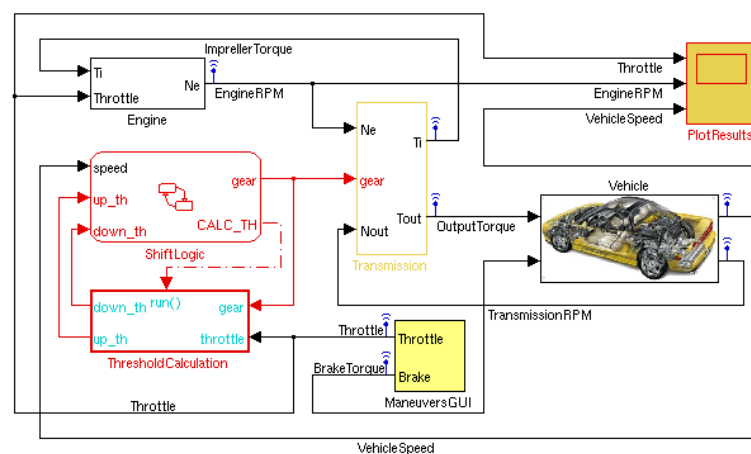


Figura 6.13: Exemplo de implementação de Autómato Híbrido

Os estados formam o objecto básico no *StateFlow* e reflectem modos de funcionamento

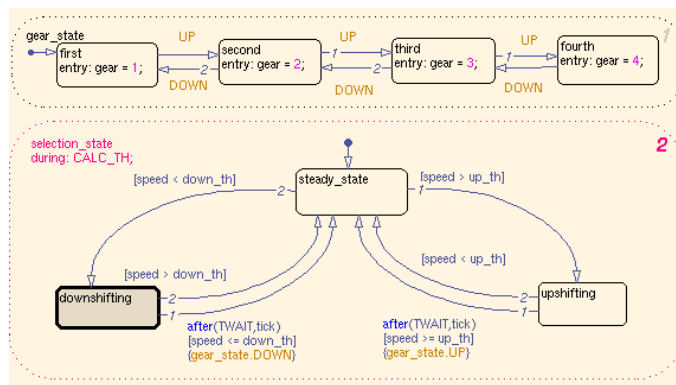
de um sistema dinâmico. Os estados podem estar activos ou inactivos. Activo significa que o *StateFlow* se encontra no modo do estado activo, no entanto vários estados podem estar activos ao mesmo tempo o que significa que o *StateFlow* permite paralelismo entre estados. Eventos e condições fazem com que os estados passem do modo activo para inactivo e vice versa. O *StateFlow* suporta também a hierarquia entre estados. Na figura 6.14 o estado *gear_state* é o estado "Pai" dos estados *first*, *second*, *third* e *fourth*. Os eventos e transições entre eles estão contidos no estado *gear_state*. Existem duas principais decomposições de estados: *OR* e *AND*. Na decomposição *OR* apenas um subestado pode estar activo ao mesmo tempo. Por sua vez na decomposição *AND* todos os subestados estão activos ao mesmo tempo dentro de um estado. No modo gráfico é possível observar nos estados algumas etiquetas de texto que denotam as propriedades dos estados que são: acções *entry*, acções *during*, acções *exit* e acções *on event*.

- **Entry** define as acções a serem tomadas quando o estado é activado;
- **During** define as acções que são tomadas quando o estado já está activo e algum evento ocorre;
- **Exit** define as acções tomadas quando o estado passa do estado activo para inactivo;
- **On event** define as acções a serem tomadas quando o estado está activo e um determinado evento ocorre.

As transições são os "saltos" entre estados. Etiquetas estão associadas às transições e podem consistir em um evento, uma condição, uma acção condicionada ou numa transição acção com a seguinte sintaxe:

evento[condição]{condição acção}\transição acção

- **Evento** especifica o evento que deve ocorrer para que a transição ocorra;
- **condição acção** especifica a expressão booleana que precisa de ver avaliada como verdade para que a transição ocorra;
- **condição acção** especifica a acção a ser executada imediatamente quando a condição for verdadeira;
- **transição acção** especifica a acção a ser executada quando o destino da transição é validado.

Figura 6.14: Exemplo de um Bloco *StateFlow*

6.8.2 Exemplo *StateFlow*

Esta secção apresenta uma possível implementação do autómato exemplo da figura 6.12 bem como os resultados. Na figura 6.15 está representado o modelo *Simulink* e o respectivo modelo em *StateFlow* da uma possível implementação do autómato da figura 6.12. O modelo *StateFlow* é constituído por dois estado e duas transições. Os estados são o estado *off* e o estado *on*, em que o estado *off* é o estado inicial. As transições também são duas e estão identificadas com as etiquetas com as condições de transição. A transição do estado *on* para *off* é feita com a condição $[\text{temp} > 21]$ e a transição do estado *off* para *on* é efectuada sempre que se validar a condição $[\text{temp} < 19]$. Na figura 6.16 é possível observar o resultado da simulação do modelo.

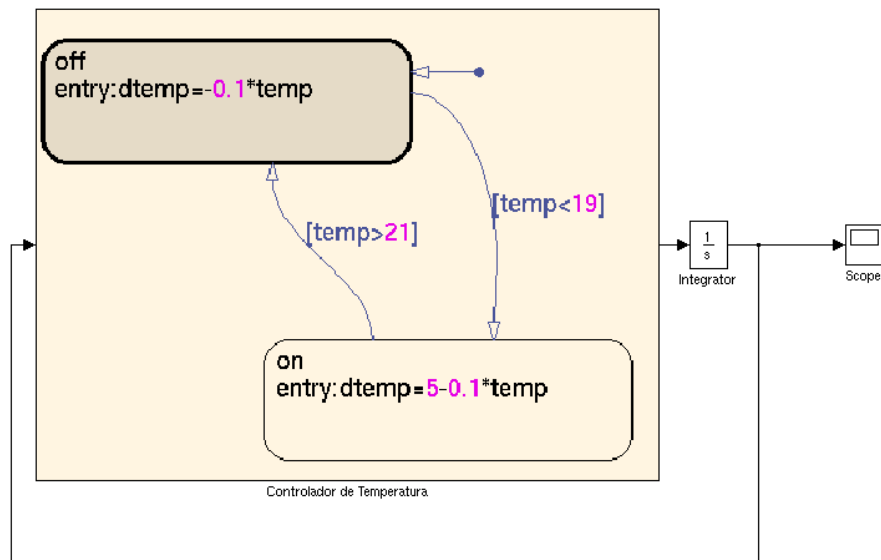


Figura 6.15: Implementação exemplo *StateFlow*

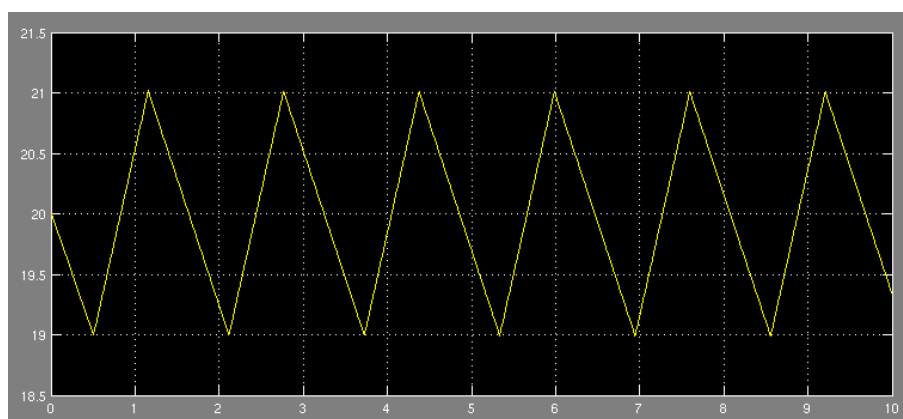


Figura 6.16: Resultado da implementação exemplo *StateFlow*

6.9 Estimação e filtro de Kalman

Estimação é definida como um método que permite determinar um único conjunto de valores para um conjunto de parâmetros (x), a partir de um conjunto de medidas (z) contaminadas com um ruído (η). A descrição matemática é feita na equação 6.11, para o caso do problema ser linear:

$$z(t) = H(t)x(t) + \eta(t) \quad (6.11)$$

onde,

- $H(t)$ representa a matriz do modelo da medição (ou matriz de observação) no instante de tempo t ;
- $\eta(t)$ representa o ruído da medida no instante de tempo t , com a matriz de densidade espectral $R(t)$.

É possível resolver a equação 6.11 como um conjunto de equações lineares de forma a obter uma estimativa do valor de x , isto se o número de observações for igual ao número de parâmetros. Quando existe um maior número de observações do que parâmetros, então o observador calcula uma solução que é ideal em algum sentido. Por exemplo, o estimador de mínimos quadrados (*Least Square Estimator*) [50] neste caso fornece uma solução que minimiza a soma dos resíduos ao quadrado. Como foi visto anteriormente o estimador calcula o estado do sistema apenas baseado no modelo de medição, como tal se o número de medidas for menor do que o número de parâmetros não é possível ao estimador calcular o estado do sistema. No entanto se o estimador utilizar informação relativa à dinâmica do sistema no cálculo não só é possível calcular o estado do sistema mas também é possível obter uma melhor estimativa dos parâmetros. A dinâmica do sistema (também conhecido como modelo do processo) pode ser representada pela equação 6.12:

$$\dot{x}(t) = F(t)x(t) + G(t)w(t) \quad (6.12)$$

onde,

- $(\dot{\cdot})$ representa derivada em ordem ao tempo
- $F(t)$ representa a matriz de dinâmica do sistema no instante de tempo t ;

- $G(t)$ representa a matriz do ruído da dinâmica do sistema no instante de tempo t ;
- $w(t)$ representa o ruído do processo, no instante de tempo t , com densidade espectral $Q(t)$

Na equação 6.12 o aparecimento de ruído torna visível o facto que o conhecimento do utilizador em relação ao comportamento do sistema é limitado. Um estimador que usa tanto o conhecimento da dinâmica do sistema(6.12) como o conhecimento da relação entre os estados e as medições (6.11) é o filtro de *Kalman*.

Em grande parte dos casos o algoritmo de estimação, neste caso o filtro de *Kalman* é implementado num sistema computacional, a forma discreta das equações 6.11 e 6.12 é mais útil, assim a forma discreta da equação 6.12 é apresentada na equação 6.13:

$$x_{k-1} = \Phi_{k+1,k}x_k + w_k \quad (6.13)$$

onde,

- $\Phi_{k+1,k}$ é a matriz de transição de estado do instante t_k para o instante t_{k+1}
- x_k representa o vector de estado do processo no instante t_k
- w_k representa o ruído do processo no instante t_k , com a matriz de covariância Q_k

A matriz de transição $\Phi_{k+1,k}$ pode ser obtida através da matriz de dinâmica do sistema na forma contínua ($F(t)$), assumindo que a matriz de dinâmica do sistema é invariante no tempo durante o intervalo de tempo de transição. Mesmo que esta suposição não seja completamente verdade, podemos minimizar os erros reduzindo o intervalo de tempo de transição. A relação entre a matriz de transição e a matriz de dinâmica pode ser expressa pela equação 6.14

$$\Phi = e^{F\Delta t} = I + F\Delta t + \frac{F\Delta t}{2!} + \dots \quad (6.14)$$

na equação anterior I representa a matriz identidade e Δt representa o intervalo de tempo de transição.

A matriz de covariância do ruído do processo (Q_t), que representa a incerteza assumida no modelo do processo, pode ser obtida através da integração da matriz de densidade espectral na forma contínua Q_t e é dada pela equação 6.15

$$Q_k = \int_{t_k}^{t_{k+1}} \Phi_{k+1,\tau} G(\tau) Q(\tau) G^T(\tau) \Phi_{k+1,\tau}^T d\tau \quad (6.15)$$

A equação do modelo de medição 6.11, na forma discreta é dada pela equação 6.16

$$z_{k+1} = H_{k+1} x_{k+1} + \eta_{k+1} \quad (6.16)$$

onde H_{k+1} representa a matriz de concepção do sistema no instante t_{k+1} , e η_{k+1} representam ruído da medida no instante t_{k+1} , com a matriz de covariância R_{k+1} .

6.9.1 Algoritmo De Filtro De Kalman

O filtro de Kalman [51] é um algoritmo recursivo que utiliza um série de etapas de previsão e actualização de forma a estimar o estado de um processo com variância mínima. Este algoritmo assume que o processo a ser estimado pode ser na forma específica da equação 6.13, na medição do processo é assumido que ocorre em tempo discreto e em concordância com a equação 6.16. Assume também que os ruídos (W_k) e (η_k), ruído do processo e ruído da medida respectivamente, são brancos e de média zero e de correlação nula.

As equações do algoritmo do filtro de Kalman estão divididas em dois grupos, as equações de previsão do estado e respectiva covariância associada e o grupo de equações de actualização do estado e da variância associada usando as medições disponíveis.

O grupo das equações de estimação baseia-se no estado actual do sistema e no modelo do sistema assumido para obter uma estimativa *à priori* do estado seguinte, tal como demonstrado nas equações 6.17 e 6.18

$$\hat{x}_{k+1}^- = \Phi_{k+1,k} x_k^+ \quad (6.17)$$

$$P_{k+1}^- = \Phi_{k+1,k} P_k^+ \Phi_{k+1,k}^T Q_k \quad (6.18)$$

onde,

- $\hat{\bullet}$ representa a quantidade estimada;
- \bullet^- representa a quantidade de previsão, antes da actualização da medição;

- \bullet^+ representa a quantidade de actualização, depois da actualização da medição;
- P_k representa a matriz de covariância associada ao vector de estado no instante t_k .

O segundo grupo de equações é a actualização dos estados previstos e estimação da covariância com as medições disponíveis, em concordância com o modelo da medida, usando para isso as equações 6.19 e 6.20.

$$\hat{x}_{k+1}^+ = x_{k+1}^- + K_{k+1}v_{k+1} \quad (6.19)$$

$$P_{k+1}^+ = (I - K_{k+1}H_{k+1})P_{k+1}^- \quad (6.20)$$

onde,

- K_{k+1} representa a matriz de ganho de *Kalman* no instante t_{k+1} ;
- v_{k+1} é a sequência de inovação no instante t_{k+1} ;

A sequência de inovação (v_{k+1}) é a diferença entre a observação actual (z_{k+1}) e a observação prevista (\hat{z}_{k+1}) e representa a quantidade de novas informações introduzidas no sistema pelas medidas actuais. É calculada pela equação 6.21

$$v_{k+1} = z_{k+1} - \hat{z}_{k+1} = z_{k+1} - H_{k+1}\hat{x}_{k+1}^- \quad (6.21)$$

A matriz de ganho de *Kalman* é o factor de ponderação, que indica o quanto da nova informação contida na sequência da inovação deve ser aceite pelo sistema. Como tal, a matriz de ganho é otimizada de forma a produzir a mínima variância, e é dada pela equação 6.22

$$K_{k+1} = P_{k+1}^- H_{k+1}^T (H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1})^{-1} \quad (6.22)$$

O algoritmo de *Kalman* é sumariado na figura 6.17

6.9.2 Filtro De *Kalman* Não Linear

Na secção anterior é pressuposto que a relação entre as medidas e os estados é linear, contudo em grande parte dos casos práticos tal relação não acontece. Por exemplo, na

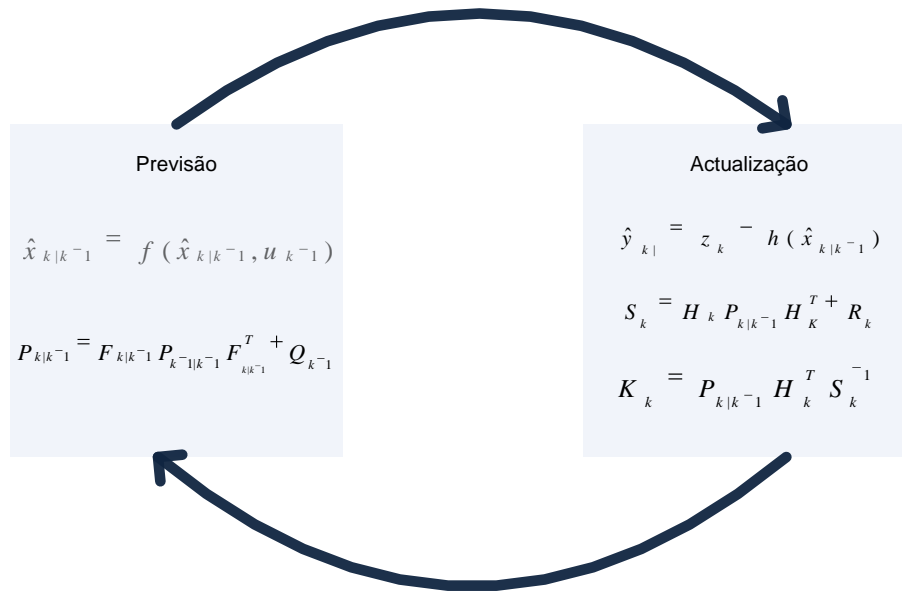


Figura 6.17: Resumo do algoritmo de *Kalman*

navegação por satélite a relação entre as medidas (*Pseudoranges* e *Doppler*) e os estados (posição, velocidade e tempo) são não lineares. Nestes casos a abordagem é linearizar o modelo numa primeira fase e só depois aplicar o filtro de *Kalman* padrão para obter o estado do sistema. Um modelo de processo e da medição não linear é apresentado nas equações 6.23 e 6.24 respectivamente.

$$x_{k+1} = f(x_k, k) + w_k \quad (6.23)$$

$$z_{k+1} = h(x_k, k+1) + \eta_{k+1} \quad (6.24)$$

Nas equações anteriores f e h são funções não lineares conhecidas. Para a linearização, é escolhida uma trajectória nominal, como é dado pela equação 6.25.

$$x_{k+1} = x_{k+1}^* + \delta x_{k+1} \quad (6.25)$$

onde \bullet^* representa o valor do vector de estado normal e δ a perturbação do valor nominal.

Assumindo que a perturbação é suficientemente pequena, a expansão em série de *Taylor* de primeira ordem das equações 6.23 e 6.24 é realizada sobre a trajectória nominal

seleccionada para obter as equações 6.26 e 6.27

$$\delta x_{k+1} = \Phi_{k+1,k} \delta x_k + w_k \quad (6.26)$$

$$\delta z_{k+1} = H_{k+1} \delta x_{k+1} + \eta_{k+1} \quad (6.27)$$

A equação 6.26 é o novo modelo de processo linear, onde o vector de estado é agora substituído pelo vector de erro de estado (perturbações). De forma análoga a equação 6.27 forma agora o novo modelo de medição linear, onde o vector de medição é substituído pela perturbação das medidas entre as medidas e as medidas previstas. As matrizes $\Phi_{k+1,k}$ e H_{k+1} , matriz de transição e matriz de modelo de medição, respectivamente, são agora função das derivadas parciais das funções não lineares respectivas com respeito ao vector de estado.

Um filtro onde a linearização é efectuada sobre o vector de estado predeterminado é designado de filtro de *Kalman* linearizado (*LKF - Linearized Kalman Filter*). Contudo, um procedimento mais comum é derivar o ponto de linearização a partir da última solução calculada (\hat{x}_k) (ao invés do vector de estado previsto, x_k). Esta abordagem é utilizada no filtro de *Kalman* extendido (*EKF - Extended Kalman Filter*). Uma vez efectuado o passo da linearização, é aplicado o filtro de *Kalman* padrão para obter as estimativas do erro do estado e os estados desejados são reconstruídos segundo a equação 6.25.

6.9.2.1 *EKF - Extended Kalman Filter*

Esta secção apresenta as equações do filtro de *Kalman* extendido. Tal como já foi explicado na secção 6.9.2 às funções f e h , modelo do processo e modelo da medição, não se lhe podem ser aplicadas a covariância directamente, para isso a cada instante as derivadas parciais são aplicadas (jacobiano). Este processo lineariza as equações para que estas possam ser aplicadas nas equações do filtro de *Kalman* directamente.

Sendo as equações 6.28 e 6.29, os modelos do processo e modelo das medições respectivamente.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \quad (6.28)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (6.29)$$

As equações do filtro de *Kalman* extendido são

Equações de previsão do estado

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \quad (6.30)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{F}_{k-1}^\top + \mathbf{Q}_{k-1} \quad (6.31)$$

Equações de actualização do estado

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}) \quad (6.32)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k \quad (6.33)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1} \quad (6.34)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \quad (6.35)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (6.36)$$

em que,

$$\mathbf{F}_{k-1} = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}} \quad (6.37)$$

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}} \quad (6.38)$$

As equações 6.30 e 6.31 representam a previsão do estado e a covariância do estado previsto, respectivamente. As equações 6.32, 6.33, 6.34, 6.35, 6.36 representam, por esta ordem: a inovação, covariância da inovação, ganho de *Kalman*, estado actualizado e covariância do estado actualizado.

Esta página foi intencionalmente deixada em branco.

Capítulo 7

Implementação e Resultados

7.1 Integração do sistema USBL

7.1.1 Hardware

A integração do sistema acústico USBL de localização relativa envolveu não só os componentes de hardware e interligação física bem como os de software. No veículo de superfície foi integrado um conjunto de hidrofones de recepção bem como o módulo de interface. O *Micron Nav Subsea Unit* acústico de localização foi instalado no sistema auxiliar de instrumentação a bordo do ROV *VideoRay*.

7.1.2 Software

O ASV possui de base uma plataforma modular de software que permite integrar novos módulos à medida que o número de sensores vai aumentando. Foi implementado um módulo de software responsável pela gestão do sensor USBL. Neste caso o módulo lê e interpreta a mensagem contendo a posição que o sensor envia.

A mensagem proveniente do sensor USBL é constituída pelos seguintes campos de informação: ID, coordenada relativa X, coordenada relativa Y, coordenada relativa Z, qualidade, tempo. Esta informação é usada posteriormente por outros processos no ASV.

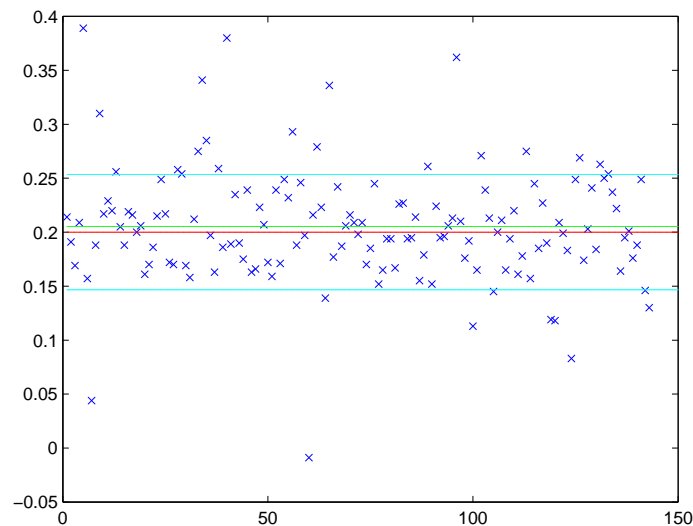
A estrutura da mensagem está descrita na tabela 7.1. Um exemplo da mensagem é apresentado de seguida “%D0030082030,1,+10,+1420,+2365,0,2,30,14342279”

Descrição	Limites
Id da unidade	000 - 016
Coordenada X (relativa)	0000000000-1000000000
Coordenada Y (relativa)	0000000000-1000000000
Coordenada Z (relativa)	0000000000-1000000000
Qualidade	0.0-3.0

Tabela 7.1: Estrutura da mensagem do sistema USBL

7.1.3 Testes Preliminares

Foram realizados alguns testes de forma a verificar as especificações do fabricante relativamente ao sistema USBL. Os testes foram realizados no tanque de testes do LSA e tinham como objectivo principal verificar a exactidão e precisão do sistema USBL. Estes testes revelaram-se inconclusivos uma vez que os dados obtidos não foram de qualidade. Sob o ponto de vista de condições de teste acústico, embora o tanque de testes tenha uma dimensão considerável ($10 \times 6 \times 5\text{m}$) este não é adequado dada a forte ocorrência de efeitos multi caminho e reflexão. O campo da mensagem qualidade enviada pelo sensor (tal como descrito anteriormente) que varia entre 0 e 3 sendo que quanto mais alto o valor melhor qualidade têm os dados apresentava valores médios de 0,2, o que se traduz em dados fracos que não permitem tirar conclusões relativamente à exactidão e precisão do sensor USBL.

Figura 7.1: Leituras USBL em x

Neste teste o *USBL Dunking Transducer* estava mergulhado a uma profundidade de 1,5 metros e o *Micron Nav Subsea Unit* a uma profundidade de 2 metros e este encontrava-se nas coordenadas (X, Y) relativamente ao *USBL Dunking Transducer* $(1, 2; 0, 2)$. Nas

figuras 7.1 e 7.2 estão representadas as leituras efectuadas pelo sensor USBL (Azul) e a distância real entre os *USBL Dunking Transducer* e o *Micron Nav Subsea Unit* está representado pela linha vermelha e a ciano estão representados os limites do desvio padrão.

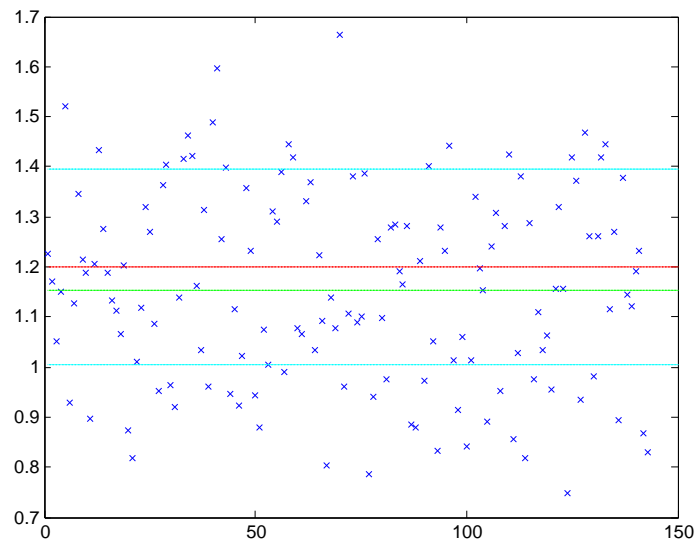


Figura 7.2: Leituras USBL em y

O sensor USBL apresenta erros de 50 cm em y que são cerca de 40% relativamente ao valor real. Outro problema encontrado nos testes no tanque foi o facto do sistema não funcionar se o *Micron Nav subsea unit* estiver a uma distância superior a 2 metros do *USBL Dunking Transducer*, isto traduz-se numa grande limitação do sistema para efectuar testes no tanque.

Foram também repetidos os testes do sistema USBL em rio com o *USBL Dunking Transducer* fixo mergulhado a cerca de 2 metros. Nestes testes o *Micron Nav Subsea Unit* foi colocado em diferentes posições x a uma profundidade de 3 metros. Os resultados obtidos são qualitativamente equivalentes aos obtidos em tanque. Verificando-se um número elevado de *pings* com factor de qualidade muito reduzido ou zero, bem como erros elevados em ambos os eixos e numerosos *outliers*. De referir que as condições de teste possíveis, num cais e com margens relativamente perto e limitações em termos de distância entre transdutores não foram os ideais.

7.2 Sistema Auxiliar de Instrumentação

Uma vez que o veículo ROV *VideoRay* possui limitações em termos de espaço interior disponível bem como de conectividade (nos conectores de interface disponível), foi implementado um sistema auxiliar de instrumentação por forma a integrar sensores não existentes no veículo bem como permitir uma interface adequada ao *Micron Nav subsea unit*. Este sistema é constituído por um módulo conversor DC/DC que é alimentado por 48 V que recebe do VideoRay e gerando as tensões de 5 V e 9 V para alimentar o microcontrolador e o INS respectivamente. O microcontrolador efectua o interface com o INS, sendo responsável pela sua inicialização (configurando-o em modo contínuo na implementação realizada), receber as mensagens através do protocolo RS-232 e fornecendo-as ao computador de bordo do ROAZ via rede CAN bus tal como descrito no fluxograma da figura 7.5.

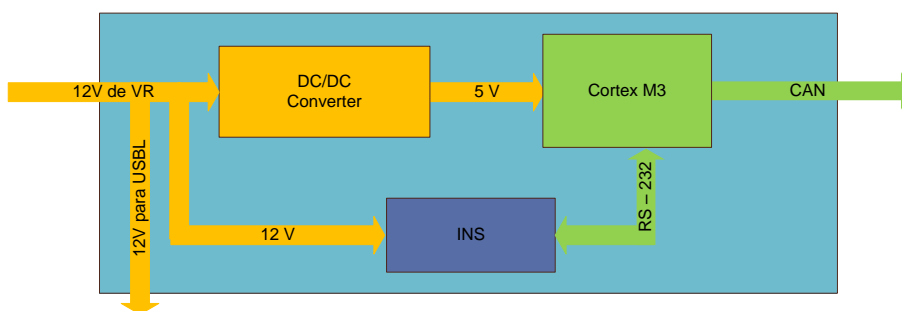


Figura 7.3: Diagrama do sistema implementado

Na figura 7.4 é possível observar o INS instalado no Sistema Auxiliar de Instrumentação 7.4(a), o pormenor da tampa com um duplo o-ring e um conector subaquático da família *SubConn® Standard Circular* 7.4(b) e o microcontrolador usado no sistema 7.4(c).

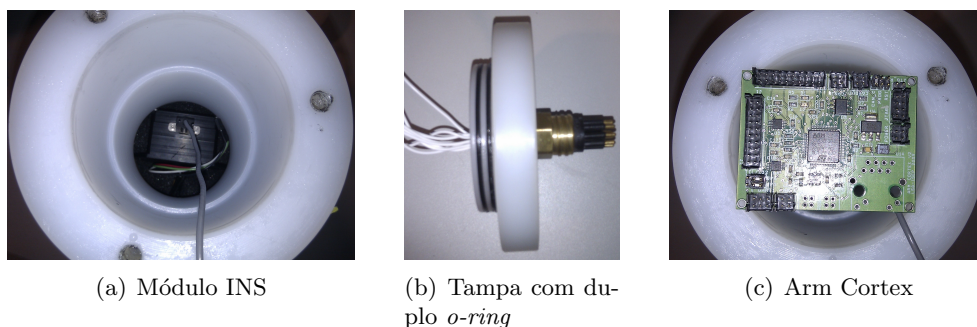


Figura 7.4: Sistema Auxiliar de Instrumentação

7.2.1 Interligações

As interligações subaquáticas são muito específicas em que a escolha dos conectores e dos cabos tem uma relevância acrescida. Assim sendo o sistema auxiliar de instrumentação dispõe de um conector *SubConn*® BH10M com as seguintes características:

- Tensão: até 600 V;
- Corrente: até 19 A;
- Pressão: até 12000 psi (equivalente a ± 822 metros de profundidade em água salgada com uma temperatura de 4°C).

Do lado do ROV *VideoRay* este possui um conector no exterior do ROV denominado de porta de acessórios. Este conector possui 9 contactos com as seguintes funções: Vídeo +, Vídeo -, aux +, aux -, Ground, 48 V+, 12 V+, grabber+ e grabber-.

7.3 INS

Foi desenvolvido software por forma a gerir os dados do sensor INS. Este software está dividido em duas partes: a primeira que consiste no software implementado no microcontrolador que permite ler os dados através do protocolo RS-232 e enviar os mesmos para o PC de bordo do ASV através do protocolo CAN, a segunda parte consiste no software que lê os dados que foram enviados por CAN e faz o seu tratamento. Na figura 7.5 está representado o software implementado no microprocessador conforme foi descrito em cima.

7.4 Simulador

O simulador implementado por forma a testar os algoritmos desenvolvidos para a coordenação dos dois veículos encontra-se representado na figura 7.6 com os seus quatro blocos. Este simulador foi implementado em *Matlab/simulink*. O simulador está dividido nos blocos: ROV, ASV, Controlador e Calculo erro. O bloco ROV está dividido em dois subblocos, o primeiro subbloco é responsável por receber e converter os dados do *joystick* em forças para que o segundo subbloco, onde estão implementadas as equações do modelo reduzido do ROV, possa simular a dinâmica do ROV. Por sua vez o bloco ASV consiste numa *s-function* que modela o ASV com um modelo cinemático simples 2D. Este bloco

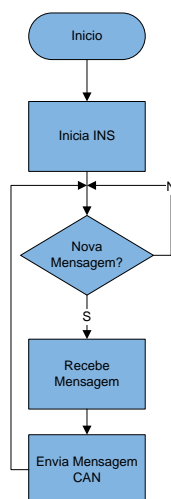


Figura 7.5: Software Micro

possui ainda um bloco que representa uma função em *Matlab* que faz o controlo do veículo de forma a seguir uma recta. O bloco controlador é constituído por um bloco *stateflow* responsável pela simulação dos estados discretos da manobra de controlo através de um diagrama de estados. O bloco calculo erro é constituído por uma função de *Matlab* que calcula a distância entre os dois veículos bem como a distância máxima a que os veículos podem estar (dependente da profundidade do ROV).

Em termos de hardware este simulador requer um *joystick* por forma a permitir a simulação da teleoperação coordenada do ROV. Quando em funcionamento o simulador apresenta dois blocos do tipo *xy graph* que permite visualizar as trajetórias de ambos os veículos, como podemos visualizar na figura 7.7.

O bloco ROV é constituído por 2 subblocos: o bloco do *joystick* e o módulo da dinâmica do ROV. No bloco principal ROV temos como entrada o sinal *enable ROV* que habilita ou desabilita o controlo do ROV e tem como saídas a posição do ROV X,Y,Z. O subbloco *joystick* utiliza o bloco que lê o *joystick*, um bloco *demux* de forma a dividir os sinais dois diferentes eixos e cada sinal do *joystick* é transformado numa força nas diferentes direcções que os propulsores permitem e essas forças são a saída do subbloco *joystick*. No subbloco dinâmica do ROV (figura 7.8) estão implementadas as equações que foram descritas na secção 5.4.1. Tem como saída a posição do ROV e como entrada as forças em X e N e a profundidade desejada.

O bloco cálculo de erro é um bloco do tipo “*embedded matlab function*”. Este bloco

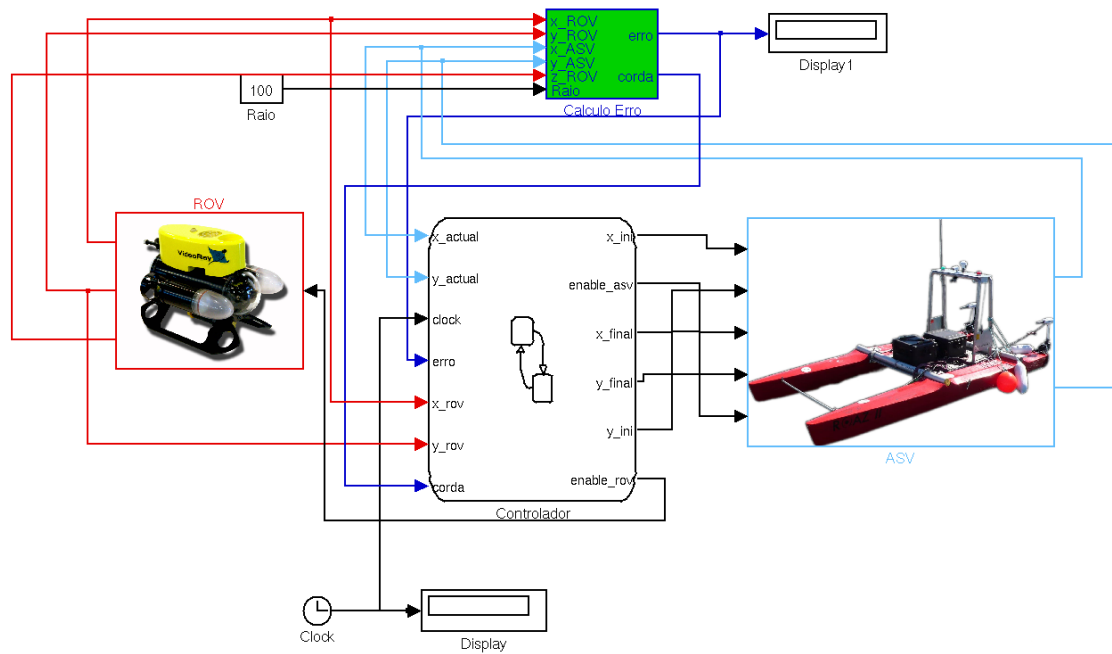


Figura 7.6: Simulador

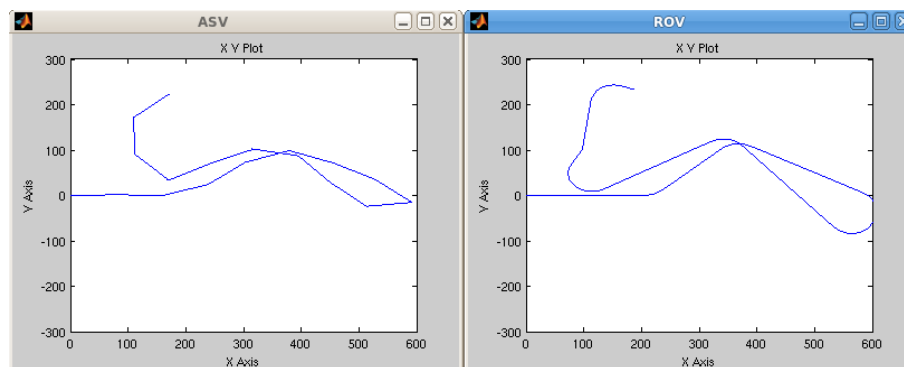


Figura 7.7: Trajetória dos veículos

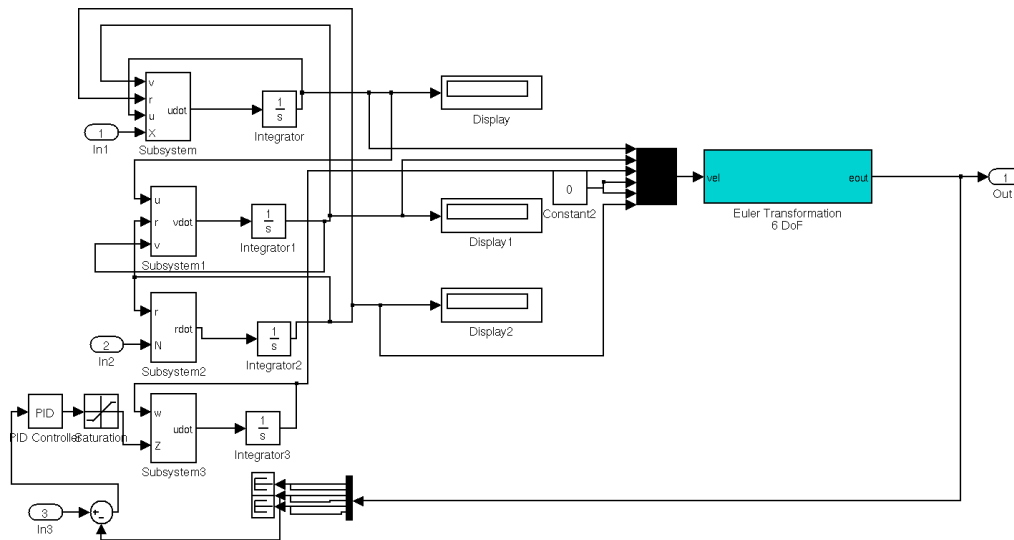


Figura 7.8: Bloco equações movimento ROV

recebe como entrada as posições actuais do ROV, ASV e o raio máximo de acção e calcula a distância no plano X,Y entre os dois bem como o erro máximo entre os dois tal como está descrito na secção 6.6.

O bloco ASV (figura 7.9) está dividido em 2 subblocos, um de controlo de trajectória e outro que simula a cinemática do veículo ASV. O bloco de controlo de trajectória é constituído por um *embedded matlab function* que recebe como parâmetros de entrada os pontos inicial e final de uma recta, o estado actual do ASV e o sinal de bloqueio do ASV, com estes parâmetros de entrada a função gera as velocidades para os motores do ASV de forma a este seguir a recta definida pelos parâmetros de entrada como está descrito em 6.5.1.1. O bloco da cinemática do ASV recebe como entrada as velocidades originárias do bloco de controlo de trajectória e tem como saída o estado do robô (x,y,θ)

O bloco controlador (figura 7.10) é responsável pela gestão da manobra de controlo implementada. Este bloco tem como parâmetros de entrada as posições do ROV, do ASV, o erro (distância entre os dois no plano x, y) bem como a corda (distância máxima admitida para a profundidade actual), a saída do bloco são os parâmetros de entrada do bloco controlo de trajectória e os sinais de controlo do ROV e ASV. Este tem como função gerir o movimento dos dois veículos, isto é, quando o ROV se afastar demasiado o ASV irá segui-lo através da trajectória calculada por o controlador.

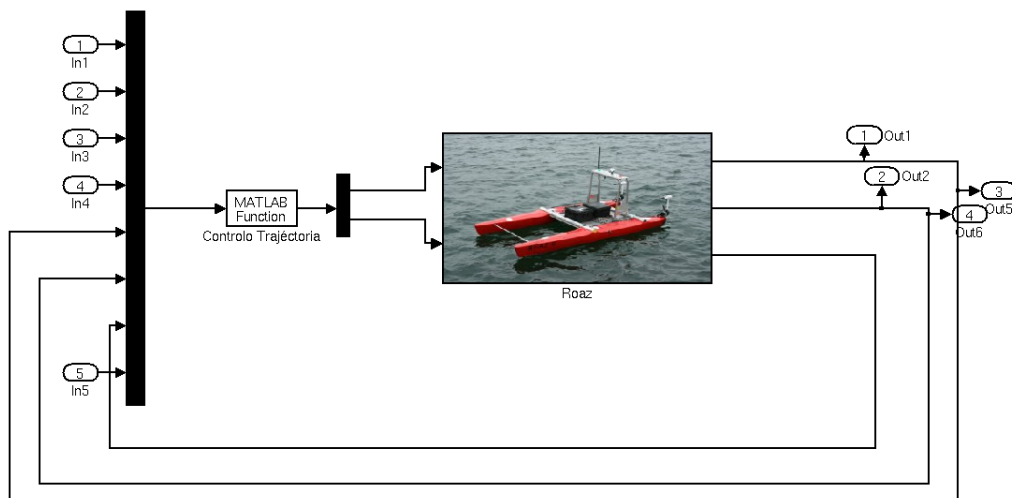


Figura 7.9: Bloco ASV

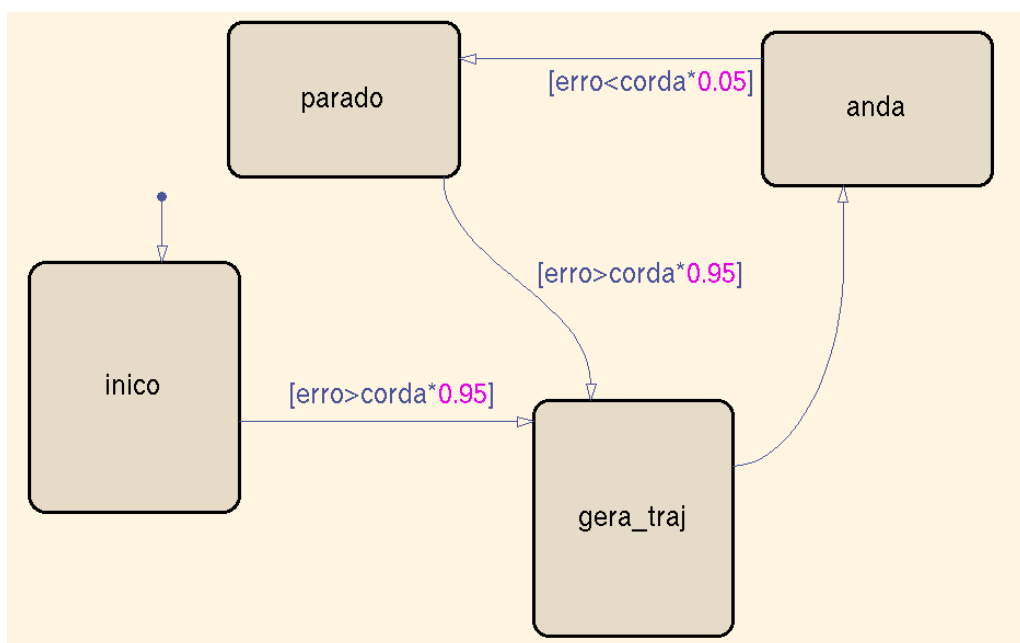


Figura 7.10: Diagrama de estados do bloco controlador

7.5 Resultados Do Simulador

7.5.1 Teleoperação

Nesta secção são apresentados resultados da simulação da manobra de teleoperação remota do ROV. Nesta simulação era pretendido simular uma manobra de teleoperação remota em que o ROV é teleoperado e o ASV sempre que a distância horizontal entre os dois veículos ultrapassar um determinado valor (dependente da profundidade do ROV) posiciona-se de forma a minimizar essa distância e assim permitir ao ROV dispor sempre de cabo. Assim na figura 7.11 está representada a trajectória descrita pelo ROV e pelo ASV na simulação efectuada. A vermelho está representada a trajectória descrita pelo ASV a verde a trajectória efectuada pelo ROV. Nesta simulação o ROV foi controlado por um *joystick*. Os pontos verdes representam os vários *waypoints* gerados cada vez que o ROV se afasta do ASV (tal como descrito na secção 6.6). Usando os pontos A e B representados na figura 7.11 como exemplo, o ASV apenas se desloca do ponto A ao ponto B quando o ROV atinge os limites definidos e que nesta figura estão representados como sendo a linha magenta.

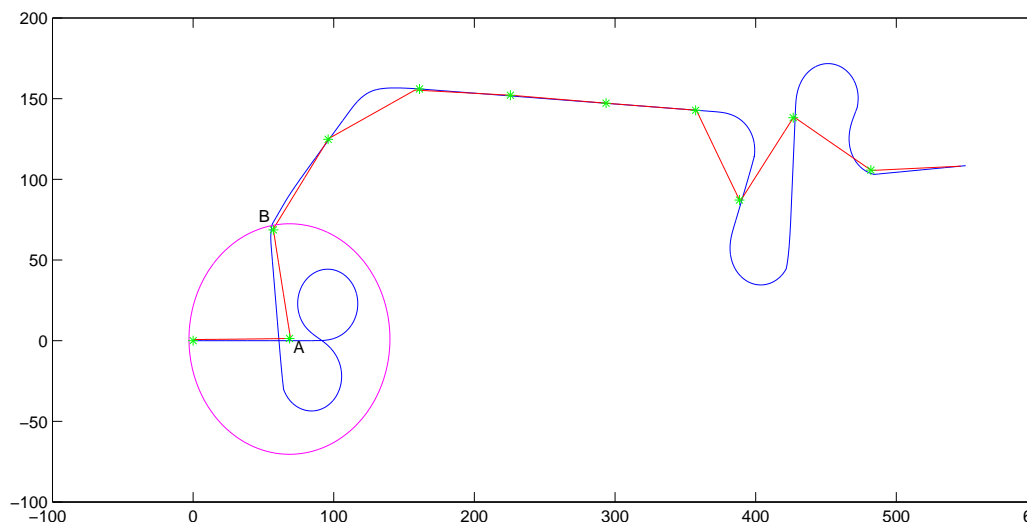


Figura 7.11: Trajectória do ASV e ROV

Na figura 7.12 podemos observar em cima a azul o erro (distância entre os dois veículos no plano horizontal) entre o ASV e ROV. As linhas verde e vermelha representam o limite de trabalho do ROV (notar que o ROV efectua uma mudança de profundidade de 10 metros para 30 metros pelo que a linha verde é válida apenas até aos 25 minutos). A linha

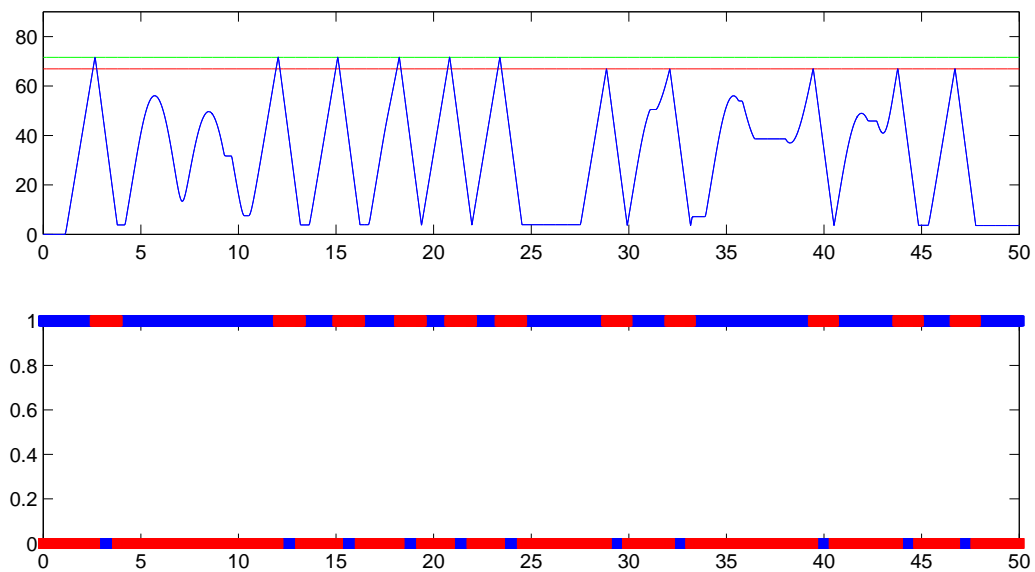


Figura 7.12: Erro e sinais de *enable*

verde é o limite para uma profundidade de dez metros e a vermelha para uma profundidade de trinta metros. Ainda na figura 7.12 em baixo podemos observar os sinais de *enable* para o ASV e ROV, assim é possível observar que sempre que o ROV está em funcionamento o ASV está parado e é também possível observar que o ASV é activado quando o erro atinge os limites (aqui representados pelas linhas verde e vermelha). Na figura a vermelho está representado o sinal de *enable* para o ASV e a azul o sinal de *enable* do ROV. Os blocos ASV e ROV interpretam estes sinais e assim os veículos são controlados apenas se o seu sinal de *enable* estiver activo.

Na figura 7.13 é possível observar a variação do limite de operação (verde) com a profundidade (azul). Como era de esperar o limite de operação diminui com o aumento da profundidade.

7.5.2 Funcionamento Autónomo

Nesta secção são apresentados os resultados do simulador mas com o ROV e o ASV em modo autónomo. Nesta simulação o ROV, ao invés de ser teleoperado, segue um conjunto de *waypoints* pré-definidos e da mesma forma que na manobra de teleoperação o ASV segue o ROV de forma a minimizar a distância entre eles. Para a realização desta simulação o simulador, tal como foi descrito em 7.4, sofreu algumas alterações. No bloco ROV o

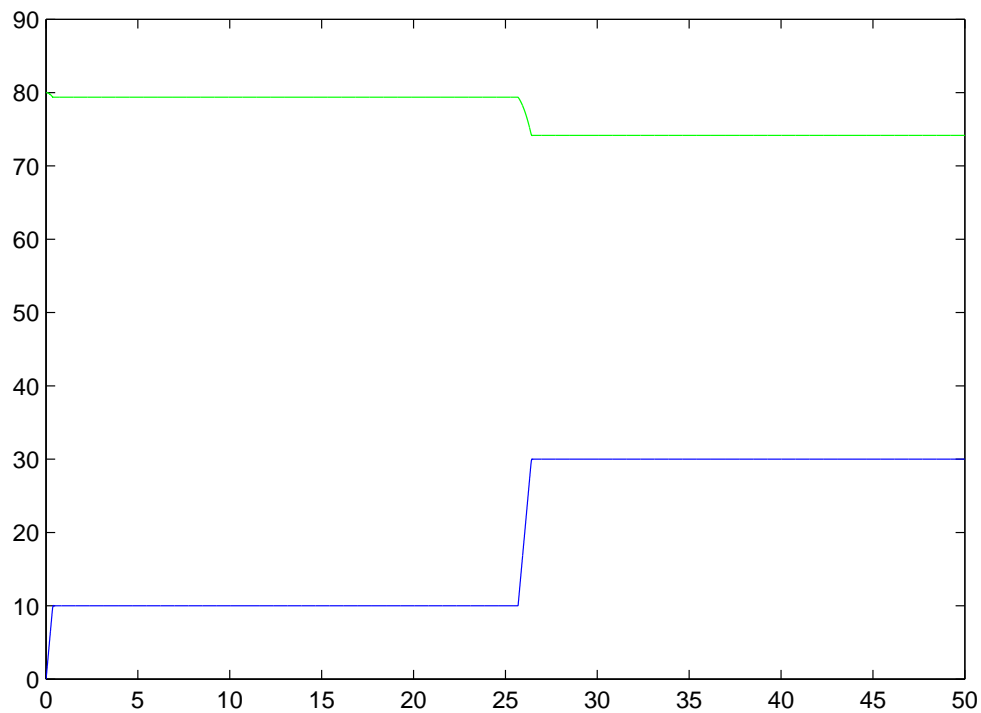


Figura 7.13: Limite de operação *vs* profundidade

joystick foi substituído por um controlador de trajetória. Este controlador é semelhante ao que foi usado no bloco ASV do simulador.

Na figura 7.14 é possível observar tal como na figura 7.11 a trajetória efectuada pelo ASV e ROV desta feita o ROV funciona no modo autónomo.

Analogamente a simulação anterior a figura 7.15 representa o erro no plano horizontal dos dois veículos e os sinais de *enable* dos mesmos. Na figura 7.16 estão representados os limites de operação e a profundidade do veículo. Nesta simulação a profundidade não foi alterada tal como na simulação anterior.

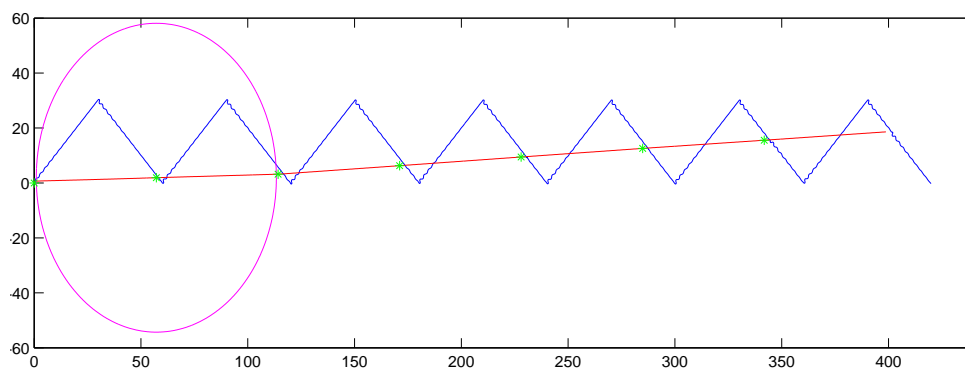
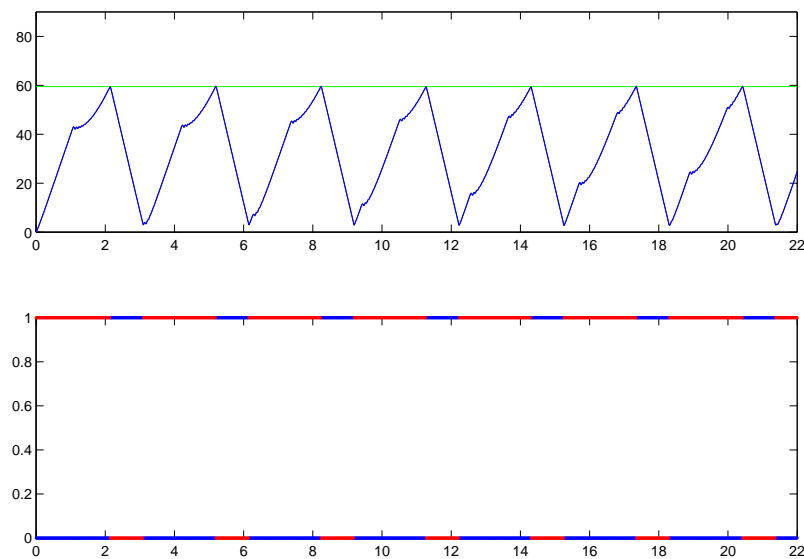
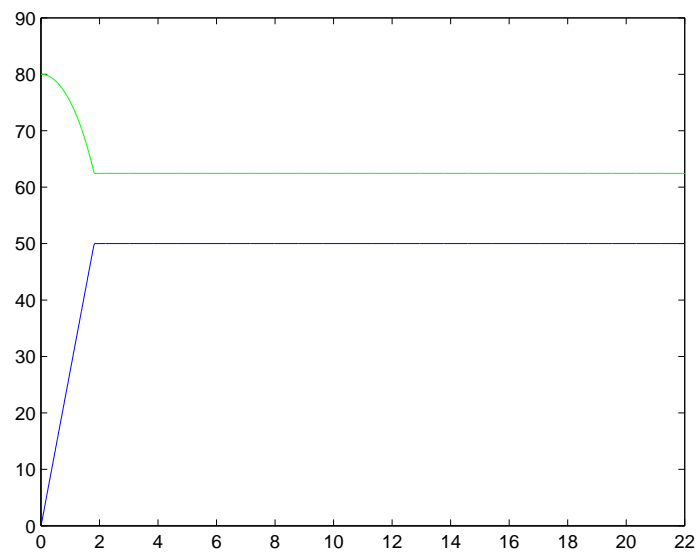


Figura 7.14: Trajetória do ASV e ROV (modo autónomo)

Figura 7.15: Erro e sinais de *enable*Figura 7.16: Limite de operação *vs* profundidade

7.5.3 Condições Da Simulação

Na primeira simulação o raio de operação foi definido como sendo oitenta metros, a profundidade do veículo até aos vinte e cinco minutos foi de dez metros depois passou para trinta metros. Na segunda simulação a profundidade do veículo foi de cinquenta metros para toda a simulação e o raio de operação definido para 80 metros.

Nas duas simulações os parâmetros do ROV usados estão descritos nas tabelas 7.2, 7.3 e 7.4 [52]

$X_{\dot{u}}$	1.94
$Y_{\dot{v}}$	6.05
$Z_{\dot{w}}$	3.95
$K_{\dot{p}}$	0.0326
$M_{\dot{q}}$	0.0175
$P_{\dot{r}}$	0.321

Tabela 7.2: Massa adicionada

X_u	2.3
Y_v	8.01
Z_w	5.81
K_p	0.0009
M_q	0.0012
P_r	0.0048

Tabela 7.3: Coeficientes de *drag* lineares

X_u	8.28
Y_v	23.69
Z_w	20.52
K_p	0.0048
M_q	0.0069
P_r	0.0089

Tabela 7.4: Coeficientes de *drag* quadráticos

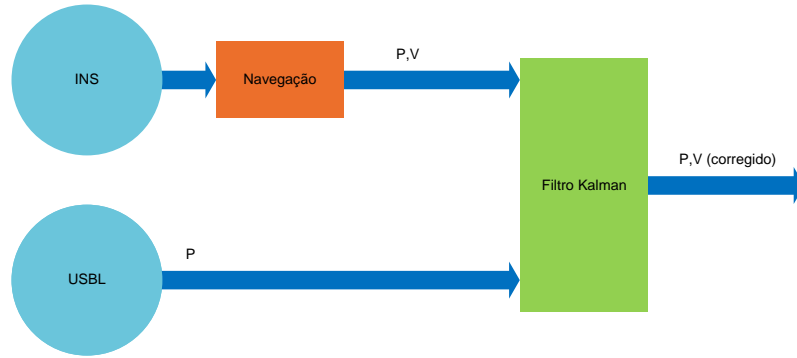
7.6 Implementação De Filtro De Localização

Foi implementado um Filtro de *Kalman* Extendido, baseado nos sensores INS e USBL. Para a localização relativa do ROV, a arquitectura do filtro é uma arquitectura *loosely coupled* [53] e está descrita na figura 7.17.

O estado do sistema é representado pelo vector X

$$X = \left[x \quad y \quad z \quad vx \quad vy \quad vz \quad bax \quad bay \quad baz \right]' \quad (7.1)$$

$[x, y, z]$ representam a posição do ROV, $[vx, vy, vz]$ representam as velocidades nas diferentes direcções do ROV e por fim $[bax, bay, baz]$ representam o erro *bias* do acelerómetro

Figura 7.17: Arquitectura Filtro *Kalman*

nas suas três componentes. O algoritmo de Filtro de *Kalman* (EKF) funciona em duas fases a previsão e *update* (secção 6.9). A equação 7.2 representa a fase de previsão que não é mais que a mecanização do sensor INS. A mecanização correspondente ao modelo de *Dead Reckoning* incorpora a informação dos sensores na qual a incerteza cresce ao longo do tempo.

$$\begin{bmatrix} P(k) \\ V(k) \\ B(k) \end{bmatrix} = \begin{bmatrix} P(k-1) + V(k) * \Delta t \\ V(k-1) + (C_n^b * a^b(k) - g^n) * \Delta t \\ B(k-1) \end{bmatrix} \quad (7.2)$$

A Matriz F (jacobiano da matriz ($F = \frac{\partial f}{\partial x}$) f (modelo do sistema) 7.2 tal como visto em 6.9.2.1) é usada no cálculo da covariância dos estados de forma a reflectir a incerteza nos estados e está representada em 7.3

$$F = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & c_n^b(1,1)\Delta t & c_n^b(1,2)\Delta t & c_n^b(1,3)\Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & c_n^b(2,1)\Delta t & c_n^b(2,2)\Delta t & c_n^b(2,3)\Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 & c_n^b(3,1)\Delta t & c_n^b(3,2)\Delta t & c_n^b(3,3)\Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.3)$$

para simplificar a leitura $\cos \cdot = c \cdot$ e $\sin \cdot = s \cdot$

$$c_n^b = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (7.4)$$

A segunda fase do filtro, a fase de *update*, é a fase na qual as medidas do sensor USBL são usadas por forma a corrigir o estado do sistema. Com o sensor USBL conseguimos medir directamente parte do estado do sistema, assim a equação de observação z fica:

$$z(k) = \begin{bmatrix} X_{usbl}(k) \\ Y_{usbl}(k) \\ Z_{usbl}(k) \end{bmatrix} \quad (7.5)$$

Na figura 7.18 podemos observar o resultado do filtro de *Kalman* a azul e a vermelho as medidas do sistema USBL. Como podemos observar o filtro descarta as medidas longe do estado real. Nos resultados das figuras 7.18, 7.19 e 7.20 o ROV estava parado nas coordenadas $(X, Y) = (-23, 5; 2)$. A figura 7.21 apresenta os dados da mecanização do INS sem a implementação do filtro de *kalman*, onde a azul estão representados os dados do INS e a vermelho os dados USBL. A figura 7.19 apresenta os resultados de forma mais

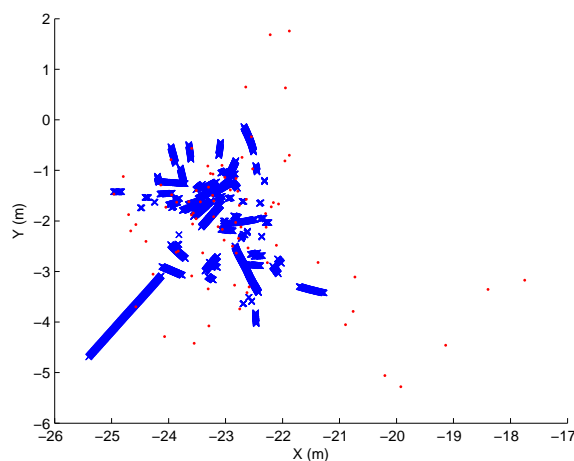


Figura 7.18: Resultado Filtro *Kalman*

detalhada no eixo x em que o valor real está representado a verde, a vermelho o resultado do filtro e a azul as medidas do sensor USBL, da mesma forma os dados para o eixo y são apresentados na figura 7.20. A figura 7.23 apresenta os resultados do filtro de *Kalman* para uma trajectória efectuada pelo ROV. Na figura 7.22 está representada a inovação

normalizada. Os limites do desvio padrão da inovação são uma medida de afinação do filtro, quando a inovação se encontra dentro dos limites do desvio padrão podemos afirmar que o filtro está afinado.

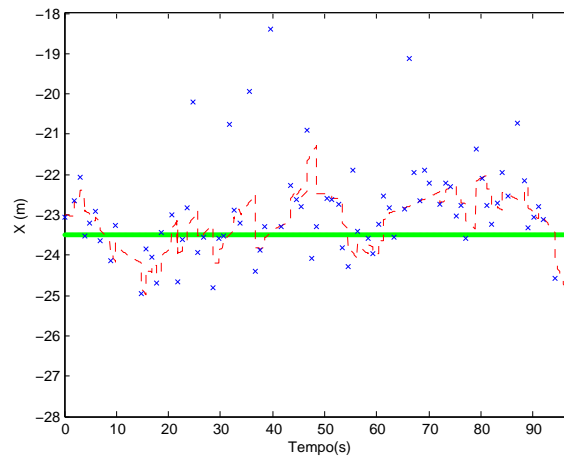


Figura 7.19: Resultado Filtro *Kalman* em y

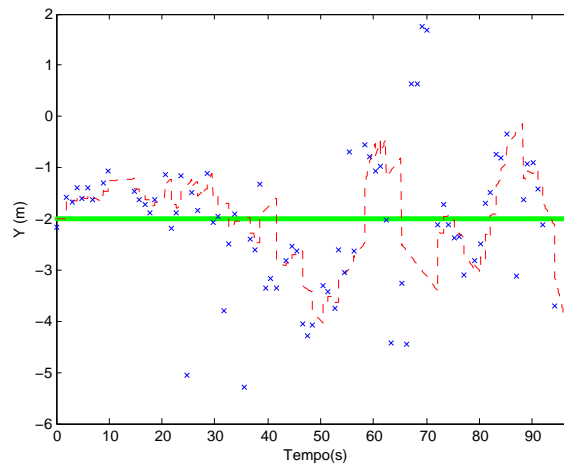
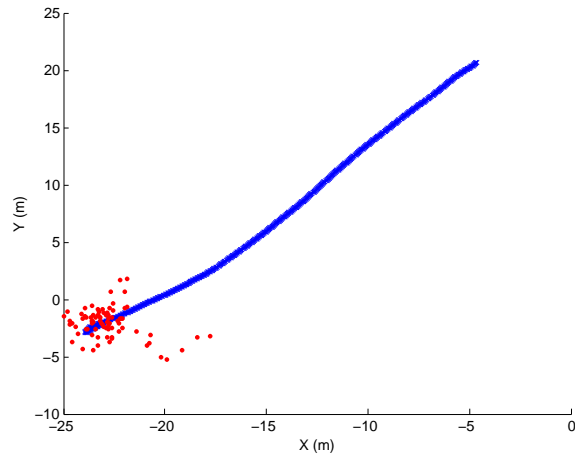
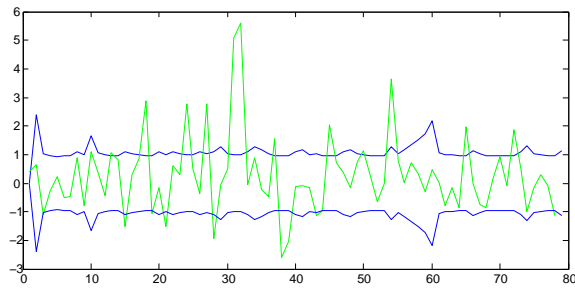
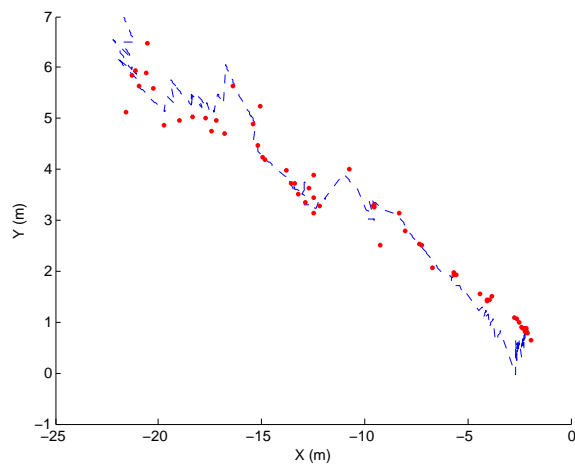


Figura 7.20: Resultado Filtro *Kalman* em x

7.7 Integração Preliminar do ASV ROV

Nesta secção é apresentada a integração preliminar do ROV no ASV em termos de *software* e *hardware*. Esta integração consiste num *setup* experimental, nas figuras 7.24 e 7.25.

Figura 7.21: INS sem filtro de *Kalman*Figura 7.22: Inovação filtro *Kalman*Figura 7.23: Resultado Filtro *Kalman*(recta)

Foi também realizada uma missão operacional no rio Douro em que foi testada a teleoperação remota do ROV.

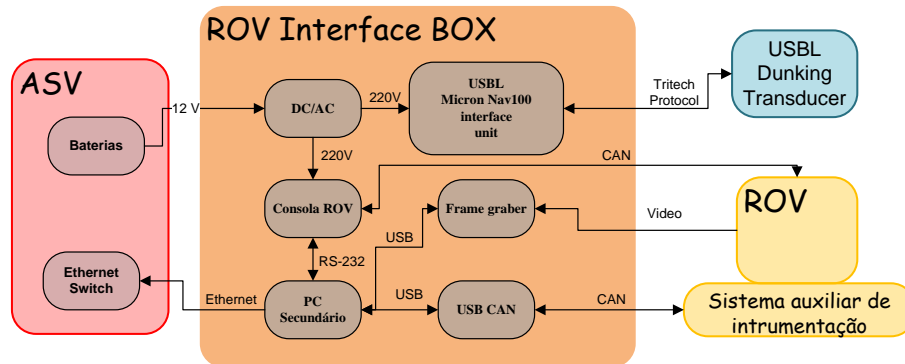


Figura 7.24: Integração em termos de *hardware*

Na figura 7.24 onde está representada a integração na vertente *hardware* é possível observar que entre o ASV e o ROV existe uma interface. Esta interface fisicamente consiste numa mala estanque (de forma familiar às restantes utilizadas no ROAZ II) onde estão todos os elementos que fazem parte da integração dos dois veículos. O veículo de superfície opera pois como plataforma de apoio ao ROV quer do ponto de vista operacional quer permitindo a georreferenciação do mesmo com recurso à posição do barco de superfície e integração do sistema USBL. Assim o ROV e o sistema USBL têm uma alimentação de 220 V AC. Para isso foi instalado um conversor DC/AC. A consola ROV comunica com o ROV através do protocolo CAN e com o PC secundário através do protocolo RS-232. De notar ainda que o PC secundário lê o vídeo de um *frame grabber* e as informações do INS a bordo do ROV através de um conversor USB-CAN.

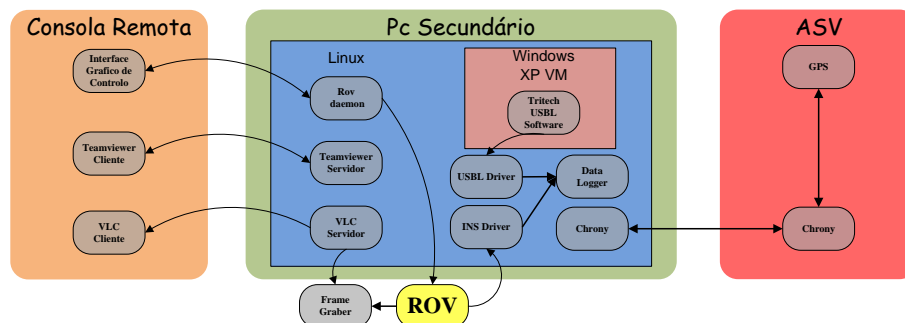


Figura 7.25: Integração em termos de *software*

Na figura 7.25 em cima temos representado um diagrama que ilustra a forma como os

diferentes processos comunicam. Os computadores existentes em terra (consola remota) e o existente a bordo do ASV que faz a interface com o ROV funcionam com o sistema operativo *Linux*. Por o sistema USBL apenas funcionar em *Windows* uma vez que o processamento de "beamforming" é efectuado externamente sendo o software dependente do sistema operativo, o PC secundário possui uma máquina virtual (instalada em *VirtualBox*) para que o software do sistema USBL funcione. No lado da consola remota existem três componentes principais, interface gráfico de controlo do ROV, TeamViewer, VLC Cliente. O processo interface gráfico de controlo do ROV comunica com o processo ROV *daemon* no lado do ASV, o processo ROV *daemon* converte as mensagens de comando da interface gráfica para comandos RS-232 para o ROV. Para controlar a máquina virtual existente na PC secundário foi usado o software *TeamViewer* que é essencialmente um software de controlo remoto gráfico. Foi efectuado um *streaming* de vídeo do PC secundário para a consola remota através do software VLC. O PC secundário tem software que faz a recolha dos dados do sistema USBL e INS, este sistema está sincronizado com o sistema do ASV (que recolhe a informação do GPS) através do serviço *chrony*. Este serviço sincroniza os relógios de máquinas distintas por forma a ser possível relacionar os dados de diferentes sensores mesmo que estes estejam em computadores diferentes. Nos testes realizados o relógio do CPU principal do ROAZ é sincronizado com pulso PPS do GPS servindo de base de tempo para a sincronização do relógio do PC secundário. Desta forma todos os relógios estão sincronizados com o tempo GPS. Verificou-se nas experiências realizadas no rio um erro de sincronismo menor do que $26 \mu\text{s}$ entre os dois relógios.

7.7.1 Cenário Operacional

Foi realizada uma missão em que foi testado a integração do ROV no ASV. A missão realizou-se em Vila Cova e teve como principal objectivo testar a manobra de teleoperação remota. Foi possível georreferenciar o ROV com os dados do GPS do ASV e do sistema USBL. Na figura 7.26 está ilustrado a azul as diferentes posições que o ASV ocupou ao longo da missão e a vermelho as posições do ROV.

Na figura 7.26 o veículo de superfície operou junto a um pilar de uma ponte no meio do rio, podendo o ROV efectuar tarefas de inspecção do mesmo.

Na figura 7.27 estão representadas as posições do ASV e ROV (azul e vermelho respectivamente), em cima de uma fotografia de satélite. Observando a imagem parece existir

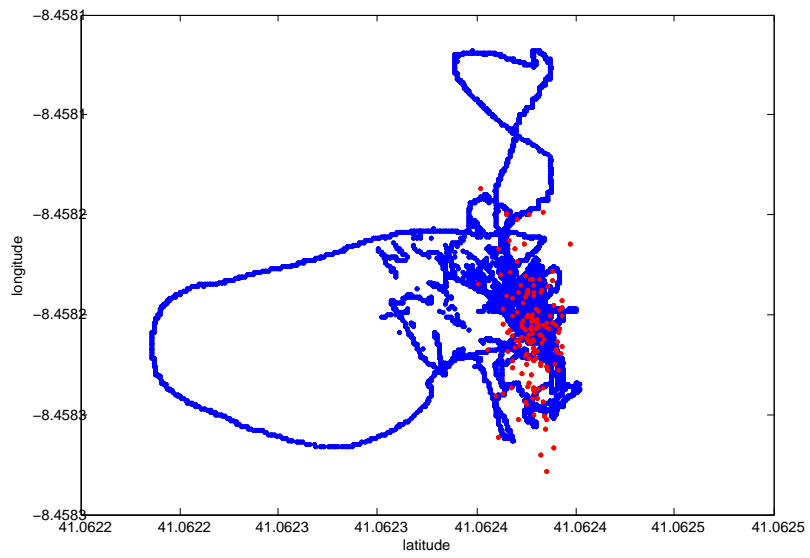


Figura 7.26: Posição do ASV e ROV no mundo



Figura 7.27: Posição do ASV e ROV no mundo(*google maps*)

um erro de georreferenciação dando a ideia de que o ASV e o ROV se encontram em cima do pilar. No entanto para além de um pequeno erro na georreferenciação da fotografia, esta foi tirada em construção pelo que nem toda estrutura visível existe agora realmente (como se pode observar (na figura 7.28)).



Figura 7.28: Roaz em missão

A missão permitiu demonstrar para além da integração do ROV com o ASV, da operação em cenário real e particularmente numa missão onde a operação conjunta é relevante tal como a inspeção do pilar de uma ponte.

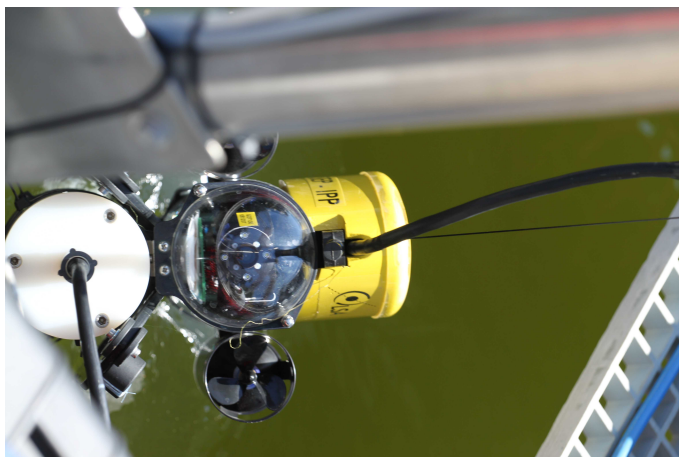


Figura 7.29: Lançamento do ROV através do ASV

Na figura 7.28 podemos observar o ASV perto do pilar que o ROV estava a inspeccionar e na figura 7.29 o ROV a ser lançado através do barco. Na figura 7.30 está representado um fotograma do vídeo recolhido na inspeção do pilar.

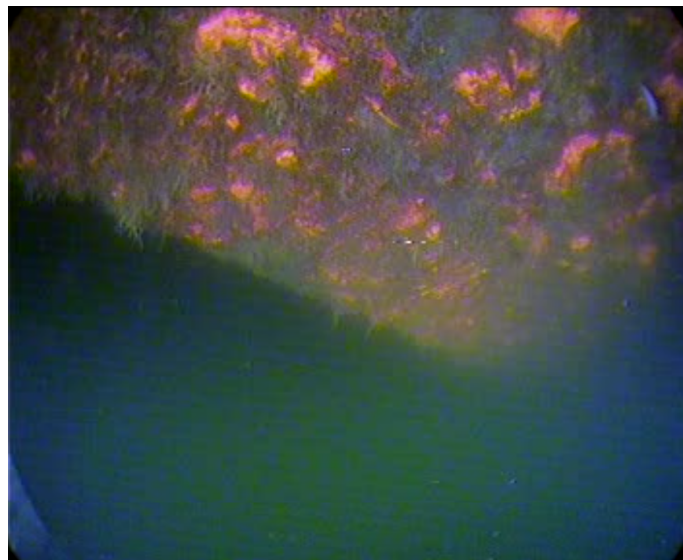


Figura 7.30: Fotograma do vídeo da inspeção

Esta página foi intencionalmente deixada em branco.

Capítulo 8

Conclusões e Trabalho Futuro

8.1 Conclusões

Neste trabalho foi abordado o problema de coordenação entre um veículo autónomo de superfície e um veículo operado remotamente, bem como descritos os requisitos que resultam da análise das aplicações em que esta solução é vantajosa. Foram também apresentados cenários operacionais típicos para o conjunto ASV/ROV.

No decorrer deste trabalho foi efectuado um estudo sobre a modelação de veículos aquáticos em 6 DOF. Neste estudo são apresentadas as equações que dão origem ao modelo dinâmico de um veículo aquático, tendo sido também apresentado um modelo reduzido e simplificado do veículo subaquático VideoRay [52]. Este modelo reduzido foi usado nas simulações efectuadas que se encontram descritas em 7.4.

Foi apresentada uma proposta de arquitectura de controlo coordenado ASV/ROV usando, como elemento para a modelação de comportamentos, um autómato híbrido. Este autómato revelou-se capaz de modelizar as estratégias de controlo necessárias à realização das manobras coordenadas entre os dois veículos. Por sua vez, estas manobras foram simuladas em ambiente *Matlab/Simulink* onde os autómatos foram testados e validados.

O projecto e implementação de um sistema de localização baseado em EKF, recorrendo a INS e USBL foi também desenvolvido. O sistema USBL revelou-se muito inferior relativamente às especificações que o fabricante apresentava, especialmente pela precisão e exactidão do sensor acústico que é muito inferior do que seria expectável. No entanto o filtro de localização cumpre a sua função de estimar a posição e velocidade do ROV bem como o *bias* dos acelerómetros do INS. A validação dos sistema USBL e do filtro de

Kalman foi possível devido à realização de diversos testes no tanque do LSA bem como em ambiente real (rio).

Devido às limitações do veículo VideoRay, descritas em 7.2, foi implementado um acessório para o ROV denominado de Sistema Auxiliar de Instrumentação, este acessório dota o ROV com a capacidade de possuir mais sensores e actuadores do que na sua forma original, neste caso os sensores INS e USBL foram adicionados ao ROV.

O veículo subaquático foi integrado no ASV em termos de software e hardware como está descrito em 7.7.1. O ROV realizou manobras de teleoperação remota, em que o ASV serve de plataforma de apoio logístico bem como de *relay* de comunicações. Foi ainda realizada uma missão onde foram testados as componentes de hardware e software da integração em cenário operacional, com o GPS do ASV e o USBL foi possível georeferenciar o ROV.

8.2 Trabalho Futuro

Em termos de trabalho futuro perspectiva-se a integração de novos modelos de ROV com o veículo de superfície. A realização de testes exaustivos em vários cenários operacionais. A implementação de um sistema de localização acústica de elevada exactidão bem como a utilização/implementação de um sistema INS com características superiores são também dois aspectos de trabalho futuro.

Outras propostas de localização do ROV vão ser estudadas e implementadas como odometria visual. Um sistema de recolha automático do ROV será um novo sistema a integrar. O veículo de superfície terá capacidade no futuro não só de servir de base a um ROV bem como a um AUV e a frotas de AUV para a realização das mais diversas missões de cooperação.

Referências

- [1] H. Ferreira, C. Almeida, A. Martins, J. Almeida, N. Dias, A. Dias, and E. Silva. Autonomous bathymetry for risk assessment with roaz robotic surface vehicle. In *Proc. OCEANS '09. OCEANS 2009-EUROPE*, pages 1–6, 2009.
- [2] E.T. Steimle, R.R. Murphy, M. Lindemuth, and M.L. Hall. Unmanned marine vehicle use at hurricanes wilma and ike. In *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, pages 1 –6, oct. 2009.
- [3] T. Maki, T. Ura, and T. Sakamaki. Map based path-planning and guidance scheme of an auv for inspection of artificial structures. In *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, pages 1 –7, oct. 2009.
- [4] Shahriar Negahdaripour and Pezhman Firoozfam. An rov stereovision system for ship-hull inspection. 31(3):551–564, 2006.
- [5] Salgado-Jimenez et al. Design of rovs for the mexican power and oil industries. In *Proc. 1st Int Applied Robotics for the Power Industry (CARPI) Conf*, pages 1–8, 2010.
- [6] J. E. Manley. Unmanned surface vehicles, 15 years of development. In *Proc. OCEANS 2008*, pages 1–4, 2008.
- [7] Universal Secure Applications. <http://www.universalsecureapplications.com/>.
- [8] Sea Robotics. <http://www.searobotics.com/>.
- [9] Alfredo Martins, Hugo Ferreira, Carlos Almeida, Hugo Silva, José Miguel Almeida, and Eduardo Silva. Roaz and roaz ii autonomous surface vehicle design and implementation. *International Lifesaving Congress*, 2007.
- [10] J. Curcio, J. Leonard, and A. Patrikalakis. Scout - a low cost autonomous surface platform for research in cooperative autonomy. In *Proc. MTS/IEEE OCEANS*, pages 725–729, 2005.
- [11] Michael Fetsch. Usv capability. In *UUXO Partnership Workshop*, 2008.
- [12] Pedro Gomes, Carlos Silvestre, Rita Cunha, and Antonio Pascoal. *A path-following controller for the delfim*. IFAC, 2006.
- [13] J. E. Manley, A. Marsh, W. Cornforth, and C. Wiseman. Evolution of the autonomous surface craft autocat. In *Proc. OCEANS 2000 MTS/IEEE Conf. and Exhibition*, volume 1, pages 403–408, 2000.

- [14] J. E. Manley. Development of the autonomous surface craft “aces”. In *Proc. MTS/IEEE OCEANS '97*, volume 2, pages 827–832, 1997.
- [15] Videoray PRO III. <http://www.videoray.com/products/10-pro-3-e>.
- [16] Seabotix. <http://www.seabotix.com/products/lbv150-4.htm>.
- [17] Phantom. <http://www.deepocean.com>.
- [18] Hercules. <http://oceanexplorer.noaa.gov/technology/subs/hercules/welcome.html>.
- [19] Triton. <http://www.perryslingsbysystems.com/rovs.html>.
- [20] Argus Bathysaurus. <http://www.argus-rs.no/xl50>.
- [21] T Curtin, J G Bellingham, J Catipovic, and D Webb. Autonomous oceanographic sampling network. *Oceanography*, 6(3):86–94, 1993.
- [22] S.R. Ramp, R.E. Davis, N.E. Leonard, I. Shulman, Y. Chao, A.R. Robinson, J. Marsden, P.F.J. Lermusiaux, D.M. Fratantoni, J.D. Paduan, F.P. Chavez, F.L. Bahr, S. Liang, W. Leslie, and Z. Li. Preparing to predict: The second autonomous ocean sampling network (aosn-ii) experiment in the monterey bay. *Deep Sea Research Part II: Topical Studies in Oceanography*, 56(3-5):68 – 86, 2009. AOSN II: The Science and Technology of an Autonomous Ocean Sampling Network.
- [23] Edward Fiorelli, Naomi Ehrich Leonard, Pradeep Bhatta, Derek A. Paley, Ralf Bachmayer, and David M. Fratantoni. Multi-auv control and adaptive sampling in monterey bay. 31(4):935–948, 2006.
- [24] J. Kalwa. The grex-project: Coordination and control of cooperating heterogeneous unmanned systems in uncertain environments. In *Proc. OCEANS '09. OCEANS 2009-EUROPE*, pages 1–9, 2009.
- [25] A. Aguiar, J. Almeida, M. Bayat, B. Cardeira, R. Cunha, A. Hausler, P. Maurya, A. Oliveira, A. Pascoal, A. Pereira, M. Rufino, L. Sebastiao, C. Silvestre, and F. Vanni. Cooperative autonomous marine vehicle motion control in the scope of the eu grex project: Theory and practice. In *Proc. OCEANS '09. OCEANS 2009-EUROPE*, pages 1–10, 2009.
- [26] A. Martins, J. M. Almeida, H. Ferreira, H. Silva, N. Dias, A. Dias, C. Almeida, and E. P. Silva. Autonomous surface vehicle docking manoeuvre with visual information. In *Proc. IEEE Int Robotics and Automation Conf*, pages 4994–4999, 2007.
- [27] J. Melo and A. Matos. Guidance and control of an asv in auv tracking operations. In *Proc. OCEANS 2008*, pages 1–7, 2008.
- [28] N. Cruz A. Matos S. Cunha S. Silva. Zarco - an autonomous craft for underwater surveys. In *7th Geomatic Week, Barcelona, Spain*, 2007.
- [29] Aníbal Matos Nuno Cruz. The mares AUV, a modular autonomous robot for environment sampling. In *Oceans'2008, Quebec, Canada*, 2008.

- [30] Gerald Gim Lee Seet Cheng Siong Chin, Micheal Wai Shing Lau Eicher Low. A robust controller design method and stability analysis of an underactuated underwater vehicle. *INTERNATIONAL JOURNAL OF APPLIED MATHEMATICS AND COMPUTER SCIENCE*, 16:345–356, 2006.
- [31] RovSim. <http://marinesimulation.com/>.
- [32] Z. Fabekovic, Z. Eskinja, and Z. Vukic. Micro rovs simulator. In *Proc. ELMAR*, pages 97–101, 2007.
- [33] L. Stutters, Honghai Liu, C. Tiltman, and D.J. Brown. Navigation technologies for autonomous underwater vehicles. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(4):581–589, july 2008.
- [34] J. Snyder. Doppler velocity log (dvl) navigation for observation-class rovs. In *OCEANS 2010*, pages 1–9, sept. 2010.
- [35] A. Matos, N. Cruz, A. Martins, and F. Lobo Pereira. Development and implementation of a low-cost lbl navigation system for an auv. In *OCEANS '99 MTS/IEEE. Riding the Crest into the 21st Century*, volume 2, pages 774–779 vol.2, 1999.
- [36] M. Morgado, P. Oliveira, C. Silvestre, and J. F. Vasconcelos. Usbl/ins tightly-coupled integration technique for underwater vehicles. In *Proc. 9th Int Information Fusion Conf*, pages 1–8, 2006.
- [37] M. Morgado, P. Oliveira, and C. Silvestre. Experimental evaluation of a usbl underwater positioning system. In *Proc. PROCEEDINGS ELMAR*, pages 485–488, 2010.
- [38] S.S. da Costa Botelho, P. Drews, G.L. Oliveira, and M. da Silva Figueiredo. Visual odometry and mapping for underwater autonomous vehicles. In *Robotics Symposium (LARS), 2009 6th Latin American*, pages 1–6, oct. 2009.
- [39] Sebastian Thrun. Robotic mapping : A survey. Technical report, CMU-CS-02-111 Technical report, Carnegie Mellon University, 2002.
- [40] Christopher Jekeli. *Inertial Navigation Systems with Geodetic Applications*. Walter de Gruyter, 2000.
- [41] David H. Titterton and John L. Weston. Strapdown inertial navigation technology. *The Institution of Electrical Engineers*, 2004.
- [42] SNAME. Nomenclature for treating the motion of a submerged body through a fluid. *Research Bulletin No. 1-5, Society of Naval Architects and Marine Engineers*, 1950.
- [43] T. I. Fossen. *Guidance and Control of Ocean Vehicles*. John Wiley and Sons Ltd, 1994.
- [44] Nuno Dias, Carlos Almeida, Hugo Ferreira, Jose Almeida, Alfredo Martins, Andre Dias, and Eduardo Silva. *Manoeuvre based mission control system for Autonomous Surface Vehicle*, pages 1–5. IEEE, 2009.
- [45] R. Skjetne, S. Moi, and T. I. Fossen. Nonlinear formation control of marine craft. In *Proc. 41st IEEE Conf. Decision and Control*, volume 2, pages 1699–1704, 2002.

- [46] Pedro Gomes, Carlos Silvestre, Rita Cunha, and Antonio Pascoal. *A path-following controller for the delfim*. IFAC, 2006.
- [47] M. Caccia, M. Bibuli, R. Bono, G. Bruzzone, and E. Spirandelli. Aluminum hull usv for coastal water and seafloor monitoring. In *Proc. OCEANS '09. OCEANS 2009-EUROPE*, pages 1–5, 2009.
- [48] J. Lygeros. Hybrid systems: Modeling, analysis and control - eecs291e lecture notes and class projects. Univ. Berkeley, 1999.
- [49] David Harel. Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, 8(3):231–274, June 1987.
- [50] Hugh Durrant-Whyte. *Introduction to Estimation and the Kalman Filter*. The University of Sydney, 2001.
- [51] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, (82 (Series D)):35–45, 1960.
- [52] Wei Wang and C. M. Clark. Modeling and simulation of the videoray pro III underwater vehicle. In *Proc. OCEANS 2006 - Asia Pacific*, pages 1–7, 2006.
- [53] Sérgio Mário Mota Coelho. Sistema de localização para veículo autónomo terrestre. Master's thesis, Instituto Superior de Engenharia do Porto, 2009.