



# Sistema de Posicionamento Acústico e Transmissão de Dados para Alvos Subaquáticos

**NUNO MANUEL COUTO VIANA**

novembro de 2019

Instituto Superior de Engenharia do Porto

# Sistema de Posicionamento Acústico e Transmissão de Dados para Alvos Subaquáticos

Nuno Manuel Couto Viana



Instituto Superior de  
**Engenharia** do Porto

Mestrado em Engenharia Eletrotécnica e de Computadores

Orientador: Alfredo Manuel Oliveira Martins  
Coorientador: José Miguel Soares de Almeida

22 de Novembro de 2019



# Abstract

With the increasing usage of unmanned underwater vehicles, the need to develop the different systems associated with them is also growing. One of the most essential systems for an unmanned vehicle is its positioning system. Due to the inherent characteristics of the medium, the most common positioning method in aquatic environments involves the use of acoustic signals that use time-of-flight measurements to locate a target, referencing it to a set of sensors.

This dissertation presents an acoustic positioning system for underwater vehicles based on the measurement of arrival delays of three acoustic signals originating from emitters with known positions on the surface, providing data transmission by periodically sending a modulated signal with position information. This system includes several components. Firstly, there are three stationary emitters on the surface responsible for the periodic sending of modulated acoustic signals. Then, there is a receiver and an emitter on the target, the former being responsible for the acquisition of acoustic signals and the latter for sending an acoustic signal with the coded position. Finally, there's a receiver on the surface responsible for capturing the signal with the coded position.

The developed detection, coding and decoding algorithms are presented, as well as the hyperbolic positioning algorithm, the latter implemented with a Kalman filter for the position estimation. In addition, the development of hardware to enable the sending and reception of the acoustic signals is also analyzed.

To validate the developed system, results obtained in field tests and in a simulation environment are presented.

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

# Resumo

Com o crescente uso de veículos subaquáticos não tripulados, cresce também a necessidade de desenvolver os diferentes sistemas a eles associados. Um dos sistemas mais importantes para um veículo não tripulado é o seu sistema de posicionamento. Devido às características inerentes ao meio, o método de posicionamento mais comum em ambientes aquáticos envolve o uso de sinais acústicos que usam medições de tempo de voo para localizar um alvo, referenciando-o a um conjunto de sensores.

Esta dissertação apresenta um sistema de posicionamento acústico para veículos subaquáticos com base na medição das diferenças de chegada de três sinais acústicos, originários de emissores com posições conhecidas na superfície, sistema esse que possibilita também a transmissão de dados através do envio periódico de um sinal modulado com informação da última posição estimada para o alvo. Na superfície, três emissores estacionários são responsáveis pelo envio periódico de sinais acústicos modulados. O alvo possui um recetor e um emissor, sendo o recetor responsável pela aquisição dos sinais acústicos e o emissor pelo envio de um sinal acústico modulado com a posição. Por fim, um receptor na superfície é responsável por capturar o sinal com a posição codificada.

São aqui apresentados os algoritmos de detecção, codificação e decodificação desenvolvidos, assim como o algoritmo de posicionamento hiperbólico usado, implementado com um filtro Kalman para possibilitar a estimação da posição. Além disso, é analisado o desenvolvimento de hardware para permitir o envio e a recepção dos sinais acústicos.

Para validar o sistema desenvolvido, são apresentados os resultados obtidos em testes de campo e num ambiente de simulação.

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

# Agradecimentos

Um sincero obrigado a todos aqueles que me ajudaram e que tornaram mais fácil esta dissertação. Um agradecimento especial aos engenheiros Alfredo Martins e José Almeida, não esquecendo também os restantes professores e investigadores do laboratório, que sempre estiveram disponíveis; aos colegas de mestrado, que contribuíram para a ultrapassagem de muitos obstáculos; aos meus pais e família, por estarem sempre do meu lado; e à Sara, por ser o meu apoio constante e por tentar sempre perceber o que me preocupava e como me poderia ajudar.

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>17</b>
1.1	Enquadramento e Motivação . . . . .	17
1.2	Requisitos do Sistema . . . . .	18
1.3	Objetivos . . . . .	18
1.4	Estrutura do Relatório . . . . .	19
<b>2</b>	<b>Estado da Arte</b>	<b>21</b>
2.1	Sistemas de Posicionamento Acústico . . . . .	21
2.2	Algoritmos de Posicionamento Hiperbólico . . . . .	26
2.3	Modulação Acústica . . . . .	27
<b>3</b>	<b>Comunicação Acústica</b>	<b>31</b>
3.1	Velocidade de Propagação . . . . .	32
3.2	Atenuação e Ruído . . . . .	33
3.3	Reflexão e Refração . . . . .	35
3.4	Efeito de Doppler . . . . .	37
<b>4</b>	<b>Formulação do Problema</b>	<b>39</b>
4.1	Arquitetura do Sistema . . . . .	42
<b>5</b>	<b>Deteção e Codificação Acústica</b>	<b>43</b>
5.1	MYTAG . . . . .	43
5.2	Deteção de Declive . . . . .	49
5.3	Funcionalidades de Comunicação . . . . .	52
<b>6</b>	<b>Posicionamento Hiperbólico</b>	<b>57</b>
6.1	Algoritmo Hiperbólico . . . . .	57
6.2	Filtro de Kalman . . . . .	62
<b>7</b>	<b>Emissão e Receção</b>	<b>67</b>
7.1	Hardware do Sistema de Emissão . . . . .	67
7.2	Hardware do Sistema de Receção . . . . .	70

7.3	Software do Sistema de Emissão . . . . .	77
7.4	Validação dos Sistemas . . . . .	82
7.5	Implementação em ROS . . . . .	85
<b>8</b>	<b>Resultados</b>	<b>93</b>
8.1	Testes em Laboratório . . . . .	93
8.2	Testes de Campo . . . . .	97
8.3	Simulação . . . . .	98
<b>9</b>	<b>Conclusão e Trabalho Futuro</b>	<b>109</b>
9.1	Conclusão . . . . .	109
9.2	Trabalho Futuro . . . . .	111
<b>A</b>	<b>Módulos Desenvolvidos</b>	<b>113</b>
<b>B</b>	<b>Código</b>	<b>117</b>
B.1	Código para a Geração do Sinal . . . . .	117
B.2	Código para a Conversão dos Dados . . . . .	118
<b>C</b>	<b>Testes em Laboratório - Resultados</b>	<b>119</b>

# Lista de Figuras

2.1	Sistema LBL . . . . .	22
2.2	Sistema SBL . . . . .	23
2.3	Sistema USBL . . . . .	25
2.4	Modulação BPSK . . . . .	28
2.5	Modulação OOK . . . . .	29
2.6	Modulação FSK . . . . .	29
3.1	SVP para águas oceânicas profundas . . . . .	33
3.2	Coefficiente de absorção para diferentes frequências . . . . .	34
3.3	Propagação energética de um sinal sonoro . . . . .	35
3.4	Relação entre frequência e SNR do sinal, para diferentes alcances . . . . .	36
3.5	Reflexão de um sinal acústico em ambiente subaquático . . . . .	36
3.6	Exemplo de refração de um sinal acústico . . . . .	37
4.1	Sistema de posicionamento . . . . .	41
4.2	Envio da mensagem para a superfície . . . . .	42
4.3	Arquitetura do sistema . . . . .	42
5.1	Sinal emitido por uma <i>tag</i> V7 . . . . .	44
5.2	Ampliação de um <i>ping</i> . . . . .	44
5.3	Resultados da aplicação do algoritmo de detecção de <i>pings</i> . . . . .	46
5.4	Seleção de uma amostra, indicando o instante temporal considerado para um <i>ping</i> . . . . .	46
5.5	Exemplo de reflexão de um <i>ping</i> . . . . .	49
5.6	Diferentes passos do novo algoritmo de detecção (1) . . . . .	51
5.7	Diferentes passos do novo algoritmo de detecção (2) . . . . .	51
6.1	Interseção hiperbólica . . . . .	58
6.2	Posição relativa ao referencial definido pelos recetores . . . . .	59
7.1	Montagem subtratora . . . . .	68
7.2	Relação entre sinal de entrada e sinal de saída da montagem subtratora . . . . .	70

7.3	Montagem do MAX680 . . . . .	71
7.4	Sistema emissor . . . . .	71
7.5	Montagem amplificadora não inversora . . . . .	72
7.6	Sinal de entrada e de saída da montagem amplificadora não inversora . . . . .	73
7.7	Montagem somadora . . . . .	74
7.8	Sinal de entrada e de saída da montagem somadora inversora . . . . .	75
7.9	Montagem inversora . . . . .	75
7.10	Sinal de entrada e de saída da montagem inversora . . . . .	76
7.11	Circuito de condicionamento de sinal . . . . .	76
7.12	LM27762 . . . . .	78
7.13	Sistema recetor . . . . .	78
7.14	Algoritmo da placa de <i>trigger</i> . . . . .	80
7.15	Algoritmo do sistema de emissão das estações . . . . .	81
7.16	Algoritmo do sistema de emissão do alvo . . . . .	82
7.17	Desfasamento entre os sinais de <i>trigger</i> gerados . . . . .	83
7.18	Periodicidade dos sinais de <i>trigger</i> gerados . . . . .	83
7.19	Sinais gerados pelos três DAC . . . . .	84
7.20	Desfasamentos dos sinais gerados pelos três DAC . . . . .	85
7.21	Sinal gerado com a mensagem codificada . . . . .	85
7.22	Amplificação do sinal gerado . . . . .	86
7.23	Ping na saída do hidrofone . . . . .	86
7.24	Ping na saída do módulo de condicionamento de sinal . . . . .	87
7.25	Nós de ROS utilizados . . . . .	88
7.26	Algoritmo geral de receção . . . . .	90
7.27	Algoritmo de calibração . . . . .	91
8.1	Testes no tanque do laboratório . . . . .	94
8.2	Disposição dos testes no tanque do laboratório . . . . .	94
8.3	Mensagem codificada adquirida pelo recetor . . . . .	96
8.4	Teste de distância . . . . .	97
8.5	Exemplo de trajetória gerada em simulação . . . . .	99
8.6	Exemplo da aplicação do algoritmo para recriação da trajetória gerada . . . . .	100
8.7	Exemplo do desempenho do algoritmo quando erros são associados às medições . . . . .	101
8.8	<i>Baseline</i> usada para o teste aos erros de posicionamento . . . . .	101
8.9	Diferença entre posição real e posição estimada . . . . .	102
8.10	Histograma dos erros de posicionamento . . . . .	103
8.11	Teste à consistência dos valores da inovação . . . . .	105
8.12	Teste à tendência do filtro . . . . .	105
8.13	Teste ao ruído branco da inovação . . . . .	106

8.14	Diferença para a posição real com e sem filtro . . . . .	106
8.15	Diferença para a posição real com e sem filtro (com falhas de leitura) . . . . .	107
A.1	Módulo de alimentação para o sistema de recepção . . . . .	113
A.2	Módulo de condicionamento de sinal para o sistema de emissão . . . . .	114
A.3	Módulo de condicionamento de sinal para o sistema de recepção . . . . .	115

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

# Lista de Tabelas

5.1	Intervalos entre <i>pings</i> nas <i>tags</i> V7 . . . . .	45
5.2	Intervalos entre <i>pings</i> dos sinais das estações . . . . .	52
5.3	Número de <i>bits</i> atribuídos a cada variável . . . . .	54
8.1	Posição (em metros) dos hidrofones no tanque . . . . .	95
8.2	Parâmetros configuráveis na simulação . . . . .	98
C.1	Resultados das posições 1 e 3 para a primeira configuração (34 medições) . . . . .	119
C.2	Resultados das posições 2 e 4 para a primeira configuração (34 medições) . . . . .	119
C.3	Resultados da posição 5 para a primeira configuração (16 medições) . . . . .	119
C.4	Resultados da posição 6 para a primeira configuração (20 medições) . . . . .	119
C.5	Resultados da posição 7 para a primeira configuração (18 medições) . . . . .	120
C.6	Resultados das posições 1 e 3 para a segunda configuração (28 medições) . . . . .	120
C.7	Resultados das posições 2 e 4 para a segunda configuração (31 medições) . . . . .	120
C.8	Resultados da posição 5 para a segunda configuração (21 medições) . . . . .	120
C.9	Resultados da posição 6 para a segunda configuração (22 medições) . . . . .	120
C.10	Resultados da posição 7 para a segunda configuração (25 medições) . . . . .	120

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

# Lista de Abreviaturas

ADC	Analog-to-Digital Converter
ARM	Advanced RISC Machine
ASK	Amplitude Shift Keying
AUV	Autonomous Underwater Vehicle
BASK	Binary Amplitude Shift Keying
BFSK	Binary Frequency Shift Keying
DAC	Digital-to-Analog Converter
DMA	Direct Memory Access
DVL	Doppler Velocity Log
DPSK	Differential Phase-Shift Keying
DSSS	Direct-Sequence Spread Spectrum
FHSS	Frequency-Hopping Spread Spectrum
FSK	Frequency Shift Keying
GPS	Global Positioning System
INESC TEC	Instituto de Engenharia de Sistemas e Computadores Tecnologia e Ciência
ISEP	Instituto Superior de Engenharia do Porto
LBL	Long Baseline
LDO	Low-Dropout Regulator
LSA	Laboratório de Sistemas Autónomos
MEEC-SA	Mestrado em Engenharia Eletrotécnica e de Computadores - Sistemas Autónomos
MFSK	Multiple Frequency Shift Keying
OFDM	Orthogonal Frequency-Division Multiplexing
OOK	On-Off Keying
PCB	Printed-Circuit Board
PPM	Pulse Position Modulation

PSK	<b>Phase Shift Keying</b>
QAM	<b>Quadrature Amplitude Modulation</b>
QPSK	<b>Quadrature Phase Shift Keying</b>
RMS	<b>Root Mean Square</b>
ROS	<b>Robot Operating System</b>
ROV	<b>Remotely Operated Underwater Vehicle</b>
SBL	<b>Short Baseline</b>
SNR	<b>Signal to Noise Ratio</b>
SSBL	<b>Super Short Baseline</b>
SVP	<b>Sound Velocity Profile</b>
TDOA	<b>Time Difference of Arrival</b>
TOF	<b>Time Of Flight</b>
USART	<b>Universal Synchronous and Asynchronous Receiver-Transmitter</b>
USBL	<b>Ultra Short BaseLine</b>
UTM	<b>Universal Transverse Mercator</b>
VPS	<b>Vemco Positioning System</b>

# Capítulo 1

## Introdução

Longe vão os tempos em que a exploração do desconhecido na superfície terrestre representava uma obra de coragem, que envolvia o uso de instrumentos regulamentares e observação dos astros para navegação e posicionamento. Nos dias de hoje, existem sistemas de posicionamento por satélite que estão implementados e disponíveis para uso por milhões de dispositivos em simultâneo, e que permitem o posicionamento com exatidão na grande maioria das áreas habitadas do planeta, sendo o mais usado o *global positioning system* (GPS). Este sistema de posicionamento usa sinais eletromagnéticos que, apesar de se propagarem com grande eficiência pelo ar, são ineficazes em ambientes aquáticos [1]. Isto originou, num planeta cuja sua superfície está maioritariamente coberta de água, o desenvolvimento de outros tipos de posicionamento para que a localização debaixo de água fosse também possível.

Uma boa localização subaquática é fundamental em diversas atividades de áreas como o estudo científico, lazer, operações de engenharia e atividades militares, como por exemplo mapeamento e busca subaquática, mineração marinha, patrulhamento, estudo de *habitats* ou posicionamento de instrumentos [2]. Um dos tipos de posicionamento mais usuais em ambientes aquáticos envolve o uso de sinais acústicos e a utilização de medições de *time of flight* (TOF) para calcular a posição de um alvo em relação a um conjunto de sensores.

### 1.1 Enquadramento e Motivação

O trabalho aqui apresentado aborda o desenvolvimento de um sistema acústico de posicionamento para múltiplos alvos subaquáticos. Este trabalho satisfaz os requisitos do Mestrado em Engenharia Eletrotécnica e de Computadores - Sistemas Autónomos (MEEC-SA), tendo surgido de acordo com as necessidades atuais do Laboratório de Sistemas Autónomos (LSA), pertencente ao Instituto Superior de Engenharia do Porto (ISEP). Este laboratório, devido à sua ligação com o Instituto de Engenharia de Sistemas e Computadores Tecnologia e Ciência (INESC TEC), trabalha frequentemente no desenvolvimento

de veículos subaquáticos e em aplicações que deles necessitem. Dos vários subsistemas que constituem um veículo subaquático, um dos mais fulcrais para a sua operação é o que está responsável pelo seu posicionamento no meio. Esta sistema é ainda mais importante quando estamos perante um veículo não tripulado, como um veículo aquático autónomo, também conhecido por *autonomous underwater vehicle* (AUV), ou veículos aquáticos operados remotamente, também denominados de *remotely operated underwater vehicles* (ROV). Os veículos existentes atualmente no laboratório utilizam um sistema de posicionamento acústico comercial, o que torna interessante o estudo e o desenvolvimento de um sistema de posicionamento próprio.

Surgiu assim o projeto de desenvolvimento de um sistema de posicionamento acústico de baixo custo que servisse como alternativa aos utilizados atualmente pelo laboratório. Este trabalho surge na continuação de um estudo previamente realizado e publicado em [3]. Neste estudo, assente na deteção e identificação de sinais acústicos, o comportamento de *tags* acústicas<sup>1</sup> comercializadas pela empresa **VEMCO** foi estudado e *software* que permitia a sua identificação a partir de gravações de hidrofones foi criado.

## 1.2 Requisitos do Sistema

O sistema a ser desenvolvido terá que ser capaz de localizar um alvo subaquático, através da receção e interpretação individual de três sinais distintos oriundos de três emissores cujas posições são conhecidas. Os dados obtidos serão depois introduzidos num filtro de Kalman, onde a posição do alvo será estimada. Além disso, a informação da posição do alvo (calculada pelo mesmo), deverá ser transmitida para a superfície de modo a ser captada e descriptada por um recetor, possibilitando o reconhecimento da posição do alvo por uma arbitrária base de operações (fixa ou móvel) existente à superfície. Para tal será necessário o desenvolvimento de *hardware* e *software* que possibilite, no que diz respeito aos recetores, a interpretação em tempo real da mensagem e estimação da posição e, no que diz respeito aos emissores, o envio assim que solicitado de determinados sinais acústicos modulados.

## 1.3 Objetivos

Tendo em vista o sistema a ser desenvolvido, foram identificados os seguintes objetivos para o presente trabalho:

- Desenvolvimento de *hardware* para permitir a modulação de informação com sinais acústicos

---

<sup>1</sup>Uma *tag* acústica é um dispositivo que emite sinais acústicos de forma a serem detetados numa região de interesse. No caso presente estas são usadas para permitir a localização de peixes previamente capturados.

- Desenvolvimento de um sistema de condicionamento de sinal para os emissores acústicos
- Desenvolvimento de uma solução para permitir a sincronização dos emissores
- Criação de *software* que permita a interpretação em tempo real dos sinais oriundos dos três emissores
- Aplicação em tempo real de um algoritmo de posicionamento hiperbólico
- Implementação de um filtro de Kalman para estimação da posição
- Validação do posicionamento em tempo real em ambiente controlado
- Realização de uma caracterização estatística da exatidão e repetibilidade do posicionamento obtido pelo sistema
- Criação de um sistema que permita a modulação e desmodulação para envio e recepção de uma mensagem por via acústica

## 1.4 Estrutura do Relatório

No primeiro capítulo é feita uma breve introdução ao problema e é exposto o enquadramento do mesmo. Além disso, são também apresentados os objetivos e os requisitos do trabalho. No segundo capítulo é feita uma exposição do estado da arte relativo a sistemas de posicionamento acústico, algoritmos de posicionamento hiperbólico e métodos de modulação acústica. No terceiro capítulo são expostas e detalhadas algumas das principais características e problemas associados com a comunicação acústica. No quarto capítulo é apresentada uma formulação detalhada do problema e a arquitetura do sistema desenvolvido. No quinto capítulo é exposto o trabalho realizado no contexto do projeto MYTAG e é feita uma análise dos métodos de codificação e decodificação desenvolvidos. No sexto capítulo é abordada a implementação do algoritmo de posicionamento hiperbólico usado e a implementação do filtro de Kalman. No sétimo capítulo é exposto o *hardware* e *software* desenvolvido para os módulos de emissão e de recepção e apresentada a implementação dos algoritmos de decodificação. No oitavo capítulo são expostos os resultados dos testes realizados em laboratório, no terreno, e em ambiente de simulação. Por fim, no nono capítulo, são apresentadas as conclusões e feitas algumas propostas para trabalho futuro.

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

## Capítulo 2

# Estado da Arte

Neste capítulo são abordados diferentes sistemas e técnicas existentes, relacionados com o trabalho realizado e aqui exposto. No subcapítulo 2.1 são apresentados os sistemas de posicionamento acústico mais convencionais, assim como uma breve descrição de cada um. Posteriormente são também mencionadas variantes desses sistemas, para ilustrar a tendência atual do desenvolvimento de sistemas acústicos de posicionamento. No subcapítulo 2.2 são apresentados métodos e algoritmos de localização hiperbólica que são comumente utilizados em problemas semelhantes àquele presente neste trabalho. No subcapítulo 2.3 são expostos e detalhados métodos de comunicação acústica, habitualmente utilizados quando está presente a necessidade de troca de dados em ambientes aquáticos.

### 2.1 Sistemas de Posicionamento Acústico

Os sistemas de posicionamento acústico podem ser separados em três grupos genéricos, dependendo do comprimento da sua *baseline*, isto é, da distância entre os diferentes sensores: *Long Baseline* (LBL), para comprimentos de *baseline* superiores a 100m, *Ultra Short Baseline* (USBL), para *baselines* inferiores a 10 cm, e *Short Baseline* (SBL), para valores de *baseline* intermédios [4].

Um sistema LBL (figura 2.1) consiste num *transceiver*<sup>1</sup> acoplado a um alvo cuja posição se pretende estimar e num conjunto de pelo menos três *transponders* instalados no fundo do mar, colocados em posições conhecidas e medidas com exatidão (para combater erros associados à posição dos *transponders*, métodos como o presente em [5] podem ser implementados). A instalação dos *transponders* no fundo do mar necessita de ser realizada previamente à realização da missão, sendo para ela necessária pessoal e material especi-

---

<sup>1</sup>Neste contexto, os termos *transceiver* e *transponder* são comumente utilizados para descrever sistemas capazes de receber e emitir sinais acústicos, sendo esta diferenciação feita para distinguir entre o sistema que responde a um sinal recebido (*transponder*) e o sistema que envia o primeiro sinal e capta a resposta do *transponder*, para assim calcular o desfasamento temporal entre os sinais (*transceiver*).

alizado (mergulhadores ou veículos subaquáticos, por exemplo). O *transceiver* presente no alvo envia um sinal que, ao ser captado pelos *transponders*, é respondido com o envio de volta de um outro sinal por parte de cada um deles. Sabendo a velocidade de propagação do sinal no meio, é possível converter a diferença entre o envio do sinal por parte do *transceiver* no alvo, e a recepção de cada um dos sinais oriundos dos *transponders*, em três medidas representativas da distância do alvo a cada um dos *transponders*, colocados em posições conhecidas. Quando estas distâncias são obtidas, é calculado o ponto onde todas estas distâncias se intersejam, estimando assim a posição do alvo, referenciado ao conjunto de *transponders* presentes no fundo do mar [6]. A este método é dado o nome de trilateração [7].

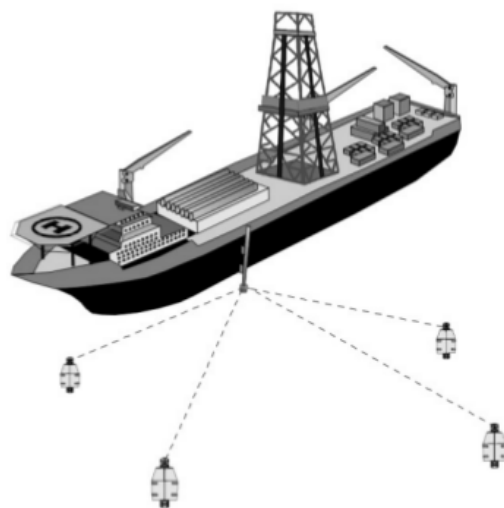


Figura 2.1: Sistema LBL [4]

Uma vez que o tempo de viagem do sinal é usado para calcular distâncias, isto trará obrigatoriamente erros associados na medição desse tempo, uma vez que o *transceiver*, acoplado ao alvo, vai estar numa posição diferente quando receber os sinais enviados pelos *transponders* em relação à posição em que se encontrava quando enviou o primeiro sinal [8], o que é especialmente notório no caso de sistemas LBL, uma vez que são normalmente utilizados em missões com uma grande área de operação, onde o alvo se poderá encontrar a uma distância considerável dos *transponders*. Desta forma, a posição estimada não vai ser representativa nem do momento em que o *transceiver* enviou o sinal, nem do momento em que este escuta as respostas dos *transponders*. Para minimizar o erro associado à medição do tempo de viagem do sinal num sistema LBL, métodos como o presente em [8] podem ser implementados.

A exatidão de um sistema LBL pode ir desde centímetros até poucos metros, para distâncias de alguns quilómetros, o que torna um sistema LBL mais exato do que um

USBL ou SBL para operações que abranjam uma área significativa. A exatidão do posicionamento está também dependente da frequência usada nos sinais do sistema LBL. Para frequências entre 8 kHz e 16 kHz, é possível obter uma exatidão entre 2 m e 5 m, enquanto que, com combinações específicas entre a distribuição dos *transponders* e a frequência utilizada, é possível obter uma exatidão inferior a 1 cm [9]. Outra das vantagens de um sistema LBL reside no facto da qualidade do posicionamento ser independente da profundidade a que este opera, o que o torna ideal para operações em águas profundas. Porém, a utilização de um sistema LBL possui algumas desvantagens. Além de ser de difícil e demorada instalação, uma vez que profissionais especializados necessitam de colocar e calibrar os *transponders* no fundo do mar, o posicionamento só é possível na área coberta pelos *transponders* instalados. Isto faz com que uma alteração da área de missão possa implicar uma alteração nas posições estabelecidas para os *transponders* [4].

Os sistemas LBL são usados não só para localizar barcos à superfície, onde outros métodos estão disponíveis, mas principalmente para localizar mergulhadores ou veículos subaquáticos. Um exemplo de um sistema LBL disponível no mercado é o HiPAP LBL da **Kongsberg Maritime** [10].

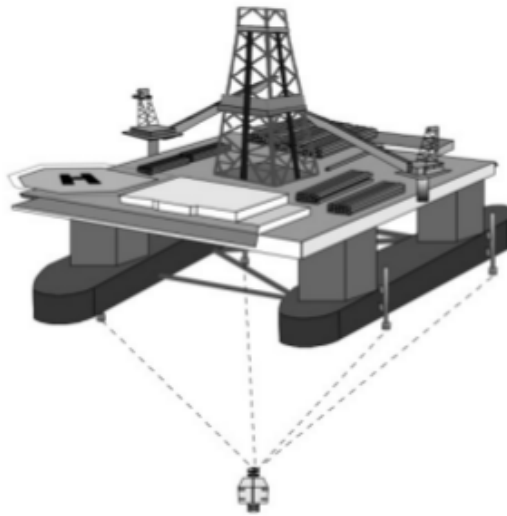


Figura 2.2: Sistema SBL [4]

Um sistema SBL (figura 2.2) tem a vantagem de, quando comparado com um sistema LBL, não necessitar de qualquer instalação de material no fundo do mar. Numa disposição SBL, três ou mais *transceivers* são colocados numa plataforma (na superfície) ou num casco de um navio enquanto que no alvo a ser posicionado é colocado um *transponder*. Um dos *transceivers* envia um sinal que é respondido pelo *transponder* com outro sinal a uma frequência distinta. Os *transceivers* presentes na estrutura perto da superfície recebem o sinal, o que possibilita a transformação do intervalo entre o envio e a chegada da resposta

em distâncias (sabendo a velocidade de propagação do sinal no meio) entre o alvo e os diferentes *transceivers* [4]. Usando um método de trilateração, a posição do alvo é obtida [7].

A exatidão do posicionamento com SBL aumenta com o aumento da *baseline*. Quando utilizado em veículos ou plataformas com dimensões consideráveis, e consequentemente com uma maior *baseline*, a exatidão do posicionamento torna-se similar à de um sistema LBL. Por outro lado, o sistema terá dificuldades em estimar corretamente a posição do alvo caso este se encontre muito próximo dos *transceivers*.

A posição estimada é referenciada ao conjunto de *transceivers* que, em oposição ao que acontecia num sistema LBL, não têm necessariamente que estar estacionários. Um vez que num sistema SBL os *transceivers* podem estar colocados num veículo em movimento, para uma estimação correta da posição do alvo será necessária a utilização de sensores adicionais, que permitam traduzir com exatidão a posição e orientação desse mesmo veículo.

Quando comparado com um sistema LBL, um sistema SBL apresenta uma complexidade e custo inferiores, dado que não é necessária nenhuma instalação no fundo do mar. Por outro lado, são necessárias *baselines* relativamente grandes para ter resultados exatos, o que pode não ser fácil de obter quando utilizado num veículo de menores dimensões [4]. Adicionalmente, a exatidão do posicionamento está dependente não só de calibrações no sistema SBL, como também da exatidão do sistema de posicionamento do veículo em que os *transceivers* se encontrem acoplados. Para colmatar erros inerentes à utilização deste tipo de sistema, métodos como o presente em [11] podem ser implementados.

Um sistema SBL é normalmente utilizado em missões que envolvam estimar a posição de um alvo subaquático em relação a um veículo de apoio na superfície ou a uma estrutura fixa, como por exemplo uma plataforma petrolífera, sendo usado preferencialmente em missões cuja área de operações é relativamente pequena, sendo um sistema LBL mais indicado para missões com uma área de operações de maior dimensão [12]. Um exemplo de um sistema SBL disponível no mercado é o SHARPS SBL da **Marine Sonic Technology** [13].

Um sistema USBL (figura 2.3), também chamado de *Super Short Baseline* (SSBL), consiste num conjunto de três ou mais *transceivers* acoplados a um navio, enquanto que um *transponder* é colocado no alvo a ser localizado. Os *transceivers* são colocados muito próximos uns dos outros, normalmente a menos de 10 cm no mesmo plano horizontal, formando duas bases ortogonais de receção [14]. É possível aumentar o número de *transceivers* para assim procurar aumentar a exatidão dos resultados, tal como demonstrado em [15], onde quatro *transceivers* foram utilizados, ou em [16], onde foi estudado o comportamento de um sistema USBL usando cinco *transceivers*.

Para a localização ser estimada, um sinal acústico é enviado por um dos *transceivers* e, após a sua deteção, um resposta é enviada de volta por parte do *transponder* presente no alvo. A diferença entre o envio do sinal e a chegada da resposta é transformada num valor

de distância usando a velocidade de propagação do sinal no meio. No entanto, ao contrário do que acontecia com os sistemas de SBL e LBL, a posição do alvo não é estimada usando trilateração. Ao invés disso, a diferença de fase do sinal entre dois recetores é registada, o que permite estimar o ângulo definido entre a parte positiva do eixo  $XX$  e o vetor de posição do alvo, projetado para o plano horizontal  $XY$ . Com a existência de terceiro recetor, ortogonal aos outros dois, é possível calcular um segundo ângulo, definido pela parte positiva do eixo  $ZZ$  e o vetor de posição do alvo. Com estes dois ângulos e com a distância estimada, a localização do alvo torna-se possível [6].

O facto de um sistema USBL não necessitar da colocação de sensores no fundo do mar torna-o mais simples e de fácil uso quando comparado com um sistema LBL. Por outro lado, tal como no caso de um sistema SBL, as medições estão referenciadas aos *transceivers* colocados numa plataforma ou num veículo na superfície, o que torna necessária a utilização de sensores adicionais que descrevam a posição e orientação dos transceivers para que seja possível a estimação da posição do alvo, estando a exatidão do posicionamento também dependente destes sensores. Um sistema USBL apresenta melhores resultados para áreas de operação pequenas e pouco profundas, apresentando-se como uma alternativa direta a um sistema SBL, tendo como principal vantagem não estar dependente do tamanho da *baseline* para obter resultados satisfatórios, o que o torna mais prático de usar quando comparado com um sistema SBL [17].

Um exemplo de um sistema USBL presente no mercado é o GAPS da **iXblue** [18].

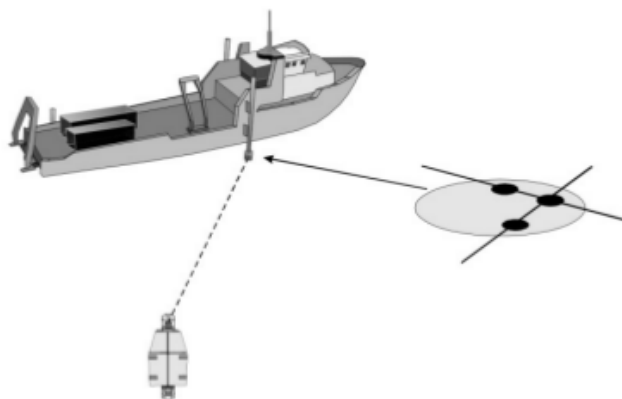


Figura 2.3: Sistema USBL [4]

É também possível combinar estes sistemas de posicionamento, resultando em sistemas com uma exatidão melhorada mas com maior complexidade. São exemplos de sistemas combinados L/USBL, L/SBL e L/S/USBL [4], [19].

Mais recentemente, novas adaptações dos sistemas anteriormente expostos têm surgido com a inversão dos mesmos.

A possibilidade de inverter um sistema USBL é vantajosa em cenários em que o alvo é muito menos ruidoso que o veículo ou plataforma de apoio, uma vez que uma melhoria em termos de *signal to noise ratio* (SNR) permite um posicionamento a maiores distâncias. Uma das principais desvantagens deste sistema reside na maior vulnerabilidade do *hardware*, dado que o sistema de *transceivers* se localiza no alvo, o que leva a custos consideráveis em caso de danos no mesmo. Por outro lado, deixar o principal elemento de um sistema USBL no alvo leva a maiores consumos energéticos em comparação com o cenário convencional, o que pode apresentar um entrave para missões que usem um AUV [19].

Outra alternativa consiste na inversão de um sistema LBL. Ao contrário do que acontece com uma disposição normal de LBL, em que os *transceivers* estão fixos no fundo do mar e como tal as suas posições são constantes ao longo do tempo, no caso de um sistema LBL invertido estes encontram-se na superfície, sendo necessária a utilização de sensores que permitam estimar a posição dos *transceivers* para que a localização do alvo seja possível. Normalmente os valores de posição dos diferentes *transceivers* são enviados para a base de apoio da missão onde, juntamente com os valores de intervalos de tempo medidos, a localização do alvo é estimada. Este sistema não permite o reconhecimento do AUV da sua própria posição estimada, uma vez que só a base da missão terá acesso a estes dados [19].

Além dos sistemas aqui expostos, outras variações estão a ser desenvolvidas e comercializadas mas, sendo o objetivo deste subcapítulo apenas fazer um retrato geral dos métodos existentes, apenas os mais relevantes foram apresentados.

## 2.2 Algoritmos de Posicionamento Hiperbólico

O posicionamento hiperbólico é realizado em duas fases. Numa primeira, é calculada a diferença de chegada de um sinal a diferentes pares de recetores, também denominada de *time difference of arrival* (TDOA). Numa segunda fase, e com o conhecimento da velocidade de propagação do sinal no meio, essa diferença de tempos é transformada numa diferença de distâncias entre o recetor e os diferentes emissores, resultando assim num conjunto de equações hiperbólicas não lineares que, quando resolvidas, permitem posicionar o recetor em relação ao conjunto de emissores [20].

A resolução do resultante sistema de equações não lineares é um processo complexo, sendo muitas vezes utilizados métodos iterativos como o presente em [21]. Este método utiliza uma estimativa inicial e melhora essa estimativa a cada iteração. Este tipo de metodologia é problemática, uma vez que para evitar que a solução fique presa a um mínimo local, é necessária uma boa estimativa inicial, o que na prática pode não ser simples. Por outro lado, não há garantias que o método iterativo convirja para uma solução [22]. A juntar ao referido, esta metodologia exige bastante em termos computacionais.

Para evitar utilizar um método iterativo na resolução das equações hiperbólicas, vários algoritmos foram desenvolvidos, sendo aqui apresentados alguns comumente utilizados.

Apresentado em [22] por Y. T. Chan, este algoritmo permite estimar não só a posição a duas dimensões de um alvo a partir de três TDOA, como também melhorar a exatidão do posicionamento quando mais de três medidas de TDOA estão disponíveis, independentemente da posição dos emissores.

No algoritmo apresentado por B. T. Fang em [23], um sistema de coordenadas é estabelecido em que um dos emissores está colocado na origem e outro numa posição no eixo  $XX$ . Se os três emissores se encontrarem no mesmo plano e seguirem esta disposição, os cálculos simplificam-se, tornando-se possível estimar duas posições possíveis para um alvo a três dimensões, caso se tenha acesso ao valor da coordenada  $ZZ$  através de um sensor externo como, por exemplo, um sensor de pressão. Fazendo uma translação dos valores de posição reais dos sensores para o novo sistema de coordenadas e aplicando o algoritmo, os valores de posição obtidos serão referenciados ao sistema de coordenadas criado. Desta forma, esses valores necessitam posteriormente de voltar a ser transladados para o sistema de coordenadas original, para que a localização desejada seja finalmente obtida. Ao contrário do algoritmo apresentado em [22], não é possível utilizar mais sensores e conseqüentemente mais valores de TDOA para aumentar a exatidão dos resultados.

Existem outros métodos de posicionamento com base em TDOA, como por exemplo o de Smith e Abel [24], o de Friedlander [25], ou o de Schau e Robinson [26], mas uma vez que a sua apresentação extensiva não é o objetivo deste documento, estes ficam aqui apenas referidos.

Devido à relativa simplicidade do algoritmo e ao facto do sistema estar limitado à coexistência de apenas três emissores, o método exposto por B. T. Fang foi o escolhido para ser implementado, tal como apresentado no subcapítulo 6.1.

## 2.3 Modulação Acústica

O envio e recepção de dados em meio aquático é fundamental para a realização de variadas tarefas. Desde sinais de controlo que traduzem comandos para veículos aquáticos, a envio de valores de telemetria ou até mesmo vídeo, esta crescente importância da transmissão de informação levou ao desenvolvimento de vários métodos e sistemas para que esta troca de dados se tornasse possível [2].

Para que a transmissão de informação se concretize, são utilizados diferentes métodos de modulação. Por modulação, entende-se o processo de variar um ou mais parâmetros de uma onda periódica (onda portadora) em função de um sinal ou mensagem que se pretende transmitir. Do lado do recetor, é necessária a existência de um processo inverso, onde o sinal modulado é transformado no sinal original. As vantagens de modular um sinal a transmitir são várias, como a melhoria da eficiência de transmissão (uma vez que

a eficiência da transmissão da mensagem está diretamente relacionada com a frequência da mesma, havendo desta forma a possibilidade de escolher uma frequência para a onda portadora conforme as necessidades), redução de ruído, ou até mesmo possibilitar a coexistência no mesmo meio de sinais com destinatários diferentes, usando portadoras com frequências distintas. Existem dois grandes tipos de modulação: modulação analógica e modulação digital [27].

Numa modulação digital, um sinal discreto é modulado com uma onda portadora periódica, permitindo transmitir um sinal digital usando um meio analógico. Este tipo de modulação traz várias vantagens em relação à modulação analógica, tais como uma maior flexibilidade (o sistema é facilmente melhorado por *software*, onde algoritmos de correção de erros ou de encriptação de mensagem podem ser adicionados ou alterados, por exemplo) e fiabilidade (uma mensagem modulada digitalmente está muito menos propensa à perda de informação durante a viagem do sinal pelo meio, como por exemplo devido a ruído e distorção do sinal). Uma das principais desvantagens assenta no facto de necessitar de sincronismo entre emissor e recetor para que os dados possam ser devidamente interpretados. Aqui serão apresentados alguns tipos de modulação digital, uma vez que esta se enquadra parcialmente nos requisitos do trabalho desenvolvido e exposto neste documento, mais concretamente no que diz respeito ao envio de dados por meio acústico.

Uma modulação *phase shift keying* (PSK) consiste na transmissão de dados mudando a fase (o ângulo inicial da senoide) do sinal de referência. Por outras palavras, a um conjunto predefinido de *bits*, é associado um valor de fase correspondente. Um das modulações mais comuns derivadas de PSK, é a *binary phase shift modulation* (BPSK), onde cada *bit* (0 ou 1) é modulado com uma mudança de fase de  $180^\circ$  [28]. O recetor analisa a fase do sinal, reconstruindo assim os dados enviados. Um exemplo de modulação BPSK pode ser observado na figura 2.4. Outras variações de PSK existem, como por exemplo, *quadrature phase shift keying* (QPSK), que possibilita a codificação de mais do que um bit por cada valor de fase, ou *differential phase shift keying* (DPSK), que altera o valor da fase atual do sinal conforme o valor do *bit* seguinte na mensagem [27].

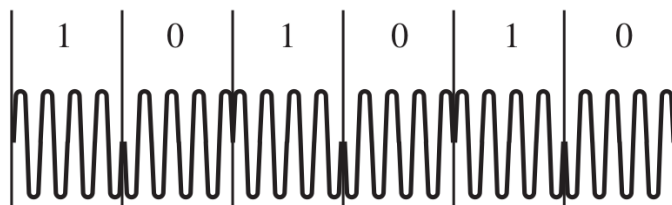


Figura 2.4: Modulação BPSK [27]

Outra tipo de modulação digital comum é a *amplitude shift keying* (ASK), que consiste em mudar a amplitude da onda portadora de acordo com a informação que se pretende transmitir, mantendo a frequência do sinal constante. Duas formas comuns de aplicar este

tipo de modulação são a *binary amplitude shift keying* (BASK) e a *on-off keying* (OOK). Na modulação BASK, um *bit* de valor 1 geralmente é modulado com amplitude superior a um *bit* de valor 0, enquanto que na modulação OOK, um *bit* de valor 0 assume na portadora uma amplitude também ela de 0, tendo o *bit* 1 um valor de amplitude arbitrário, diferente de 0 [27]. Na figura 2.5 está ilustrado um exemplo de modulação OOK.

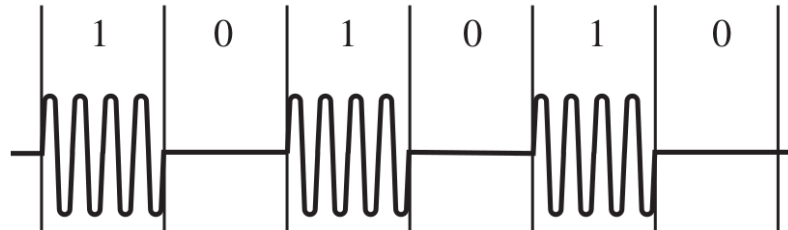


Figura 2.5: Modulação OOK [27]

Numa modulação do tipo *frequency shift keying* (FSK), a frequência da onda portadora varia conforme os *bits* presentes na informação a modular. Assim, a cada valor de *bit* é atribuído um valor de frequência correspondente, o que gera um sinal de frequência variável ao longo do tempo [27]. Um exemplo de uma variação de FSK é a *multiple frequency shift keying* (MFSK), onde o sinal é codificado recorrendo a várias frequências diferentes [29]. Na figura 2.6 está representado um exemplo de modulação FSK.

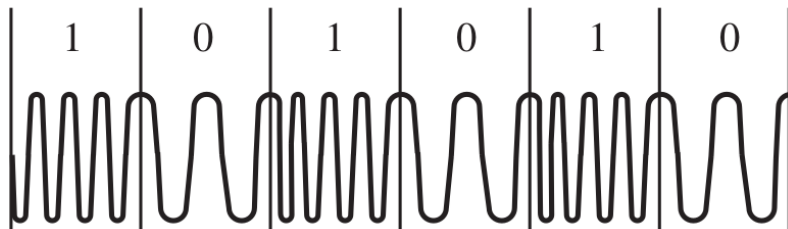


Figura 2.6: Modulação FSK [27]

Uma modulação do tipo *quadrature amplitude modulation* (QAM), permite juntar duas mensagens distintas no mesmo sinal. As duas mensagens são ambas moduladas em amplitude, mas com a mesma frequência e desfasadas em  $90^\circ$ . O facto de os sinais estarem desfasados, torna possível a sua separação e interpretação por parte de um recetor [27].

Como exemplos de outros métodos de modulação diferentes dos até aqui apresentados, podem ser mencionados o *direct sequence spread spectrum* (DSSS) [30], o *frequency-hopping spread spectrum* (FHSS) [31], o *pulse-position modulation* (PPM) e o *orthogonal frequency-division multiplexing* (OFDM) [27].

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

## Capítulo 3

# Comunicação Acústica

Tal como referido no capítulo 1, a utilização de métodos acústicos para comunicação subaquática tem crescido a par do aumento das atividades aquáticas e com a evolução tecnológica que as fomenta. No entanto, ondas acústicas não são a única forma de comunicação subaquática. Ondas de rádio de baixa frequência, de 30 Hz a 300 Hz, conseguem uma propagação a grandes distâncias em águas salgadas, mas para tal necessitam de antenas de tamanho considerável e alta potência de transmissão, enquanto que ondas de frequências menores têm uma capacidade de propagação débil [32]. Ondas óticas sofrem problemas de dispersão e atenuação, estando limitadas a distâncias curtas [33]. Por outro lado, feixes laser são bastante eficientes em termos energéticos mas necessitam de estar direcionados para o alvo para surtirem o efeito desejado [2]. Por estes e outros motivos, ondas acústicas continuam a ser a melhor solução para comunicação subaquática.

Muitas das aplicações que utilizam comunicação acústica, tanto civis como militares, exigem comunicação em tempo real com submarinos ou com veículos autónomos. Com isto em mente, o panorama atual de desenvolvimento e investigação de sistemas acústicos tem convergido maioritariamente para algoritmos de processamento de sinal, métodos de modulação mais eficientes, assim como para o estudo de diferentes técnicas de comunicação acústica. Nos diferentes sistemas já existentes, são utilizados usualmente quatro tipos de sinais: sinais de controlo (dados de navegação, comandos para veículos subaquáticos autónomos, informações de estado, entre outros), de telemetria (dados de instrumentos como hidrofones, sismógrafos, sonares e sensores químicos), de voz (comunicação entre mergulhadores, por exemplo) e de imagem (transmissão de vídeo) [2].

De uma forma geral, os sistemas de comunicação acústica estão confinados a larguras de banda inferiores àquelas usadas por comunicações terrestres com ondas de rádio. Enquanto que sinais com frequência inferior a 1 kHz possibilitam o envio de informação a distâncias superiores a 1 km, sinais com frequências a rondar os 100 kHz são muitas vezes usados em cenários com distâncias mais curtas. Este tipo de comunicação não está isento de desvantagens, uma vez que fenómenos como atenuação, refração, ou até mesmo o facto de

a velocidade do som ser relativamente baixa, o que em situações com maior dinâmica pode originar um efeito de Doppler considerável, tornam mais difícil a tarefa de comunicação acústica subaquática, quando comparada com outros meios [2].

No presente capítulo são expostas algumas dos maiores problemas associados à propagação de sinais acústicos em ambientes subaquáticos.

### 3.1 Velocidade de Propagação

Para caracterizar o comportamento de um onda acústica em ambientes subaquáticos, uma boa estimativa da velocidade de propagação do som é fundamental. Num cenário genérico de propagação de ondas sonoras no oceano, fatores como a localização, profundidade, temperatura e salinidade, influenciam diretamente a velocidade de propagação de um sinal acústico. Para uma estimativa exata da velocidade de propagação, é necessária informação exata no que toca a estes diferentes parâmetros [34].

A velocidade de propagação do som na água pode ser calculada usando vários modelos, podendo-se referir o apresentado por MacKenzie em [35] e que está presente na equação 3.1, onde  $c$  indica a velocidade de propagação do som,  $T$  a temperatura,  $S$  a salinidade e  $D$  a profundidade. As unidades desta equação e de todas as equações presentes neste documento seguem o sistema internacional de unidades.

$$c = 1448,96 + 4,591 \cdot T - 5,304 \cdot 10^{-2} \cdot T^2 + 2,374 \cdot 10^{-4} \cdot T^3 + 1,34 \cdot (S - 35) + 1,63 \cdot 10^{-2} \cdot D + 1,675 \cdot 10^{-7} \cdot D^2 - 1,02 \cdot 10^{-2} \cdot T \cdot (S - 35) - 7,139 \cdot 10^{-13} \cdot T \cdot D^3 \quad (3.1)$$

A representação da variação da velocidade de propagação do som com a profundidade é conhecida como *sound velocity profile* (SVP), estando na figura 3.1 representado um exemplo relativo a águas oceânicas profundas. Perto da superfície, a variação da velocidade de propagação com a profundidade é diminuta, uma vez que a agitação provocada pelas ondas na superfície torna a temperatura nesta zona relativamente uniforme com o aumento da profundidade. Um pouco mais fundo, numa zona apelidada de *seasonal thermocline*, a temperatura diminui de forma bastante acentuada, para depois a diminuição estabilizar em maiores profundidades, numa zona conhecida por *main thermocline*. Abaixo destas zonas, a temperatura é relativamente estável e constante, o que leva a que a velocidade de propagação do som aumente gradualmente com o aumento da profundidade e, consequentemente, da pressão. Em circunstâncias em que se verifique uma variação abrupta no SVP, como em zonas em que águas com densidades diferentes se misturam, uma mudança na direção da propagação do sinal pode também ocorrer [34].

Sabendo com exatidão a velocidade de propagação do sinal, é possível converter o

tempo necessário para que este se desloque entre emissor e recetor num valor de distância, usando a equação 3.2, em que  $t$  representa o tempo de viagem,  $d$  a distância percorrida pelo sinal e  $c$  a velocidade de propagação do som.

$$t = \frac{d}{c} \quad (3.2)$$

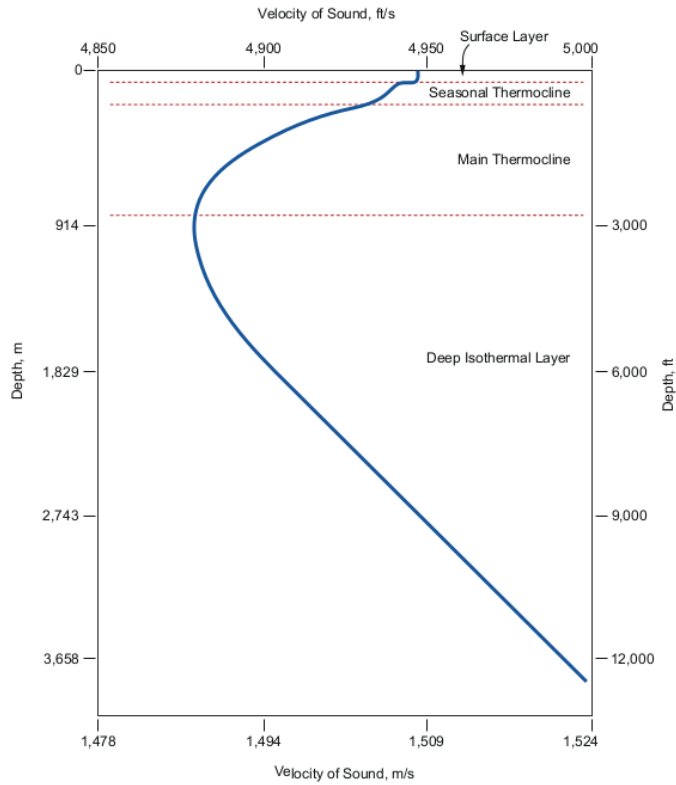


Figura 3.1: SVP para águas oceânicas profundas [34]

## 3.2 Atenuação e Ruído

O maior entrave para a propagação de um sinal acústico é a sua atenuação e consequente perda de informação. Em meios subaquáticos, a atenuação depende principalmente de dois fatores: a distância percorrida e a frequência do sinal. A importância da frequência do sinal usado é uma consequência direta da absorção, isto é, a transformação de energia presente no sinal em energia calorífica. Uma onda acústica é um distúrbio mecânico gerado pelo movimento de moléculas de água, o que leva a que calor seja gerado no momento em que as diferentes moléculas chocam umas com as outras, retirando assim energia ao sinal acústico [34]. Uma representação da absorção em função da frequência utilizada pode ser observada na figura 3.2.

Por outro lado, a distância percorrida pelo sinal influencia a atenuação presente no mesmo quando este chega a um receptor. Quando um sinal acústico é emitido num cenário ideal, este sinal irá se propagar em todas as direções, de modo uniforme. No entanto, este sinal tem energia limitada, que atinge o seu valor máximo no momento de envio. Ao mesmo tempo que o sinal se espalha no meio, a energia inicial irá ser distribuída ao longo de uma superfície esférica cada vez maior. Como a área da superfície esférica é proporcional ao quadrado da distância que o sinal viaja, de cada vez que a distância percorrida duplica, a intensidade do sinal fica quatro vezes menor [34]. Uma representação gráfica deste fenómeno pode ser observada na figura 3.3.

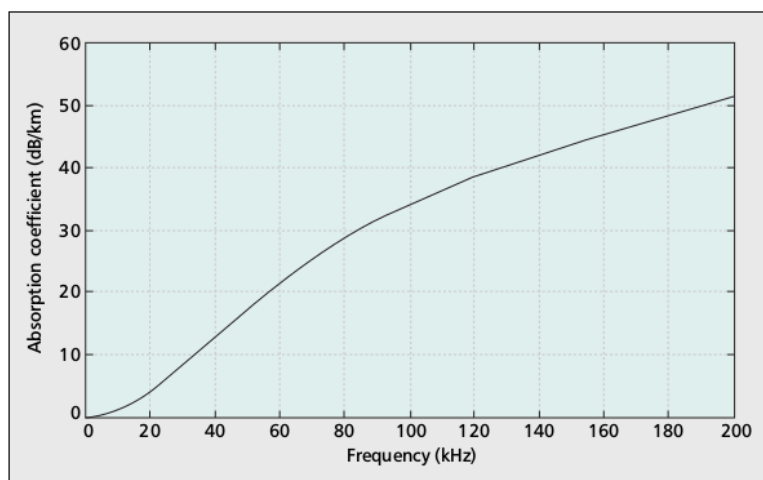


Figura 3.2: Coeficiente de absorção para diferentes frequências [36]

Outro dos condicionantes da comunicação acústica é o ruído existente no meio. O ruído ambiental vai sempre estar presente, podendo ser originado por diversos fatores tais como turbulência, rebentar das ondas ou chuva, pelo que é normalmente aproximado como ruído Gaussiano. Existirá também por vezes ruído relacionado com o local específico da operação, como por exemplo o motor de barcos ou até o quebrar do gelo em regiões polares. Ao contrário do ruído ambiental geral, este ruído específico apresenta várias componentes não Gaussianas, o que torna a sua estimativa mais difícil [36].

Por outro lado, a frequência usada na onda portadora influencia não só a atenuação do sinal como também o ruído presente no mesmo. Assim, é possível estabelecer uma relação entre a frequência usada, o alcance da mensagem, e a razão entre ruído e sinal da mensagem, ou SNR, quando captada por um receptor.

Observando a figura 3.4, é compreensível que a escolha de uma largura de banda para a onda portadora vai limitar o alcance da transmissão da mensagem, devendo o valor da frequência da portadora ser escolhido de forma a se enquadrar e otimizar as necessidades da missão [2].

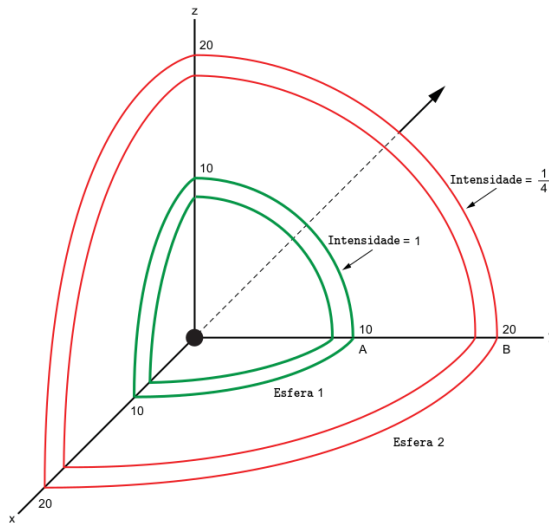


Figura 3.3: Propagação energética de um sinal sonoro (baseado em [34])

### 3.3 Reflexão e Refração

Caso o meio apresente fronteiras, tais como a superfície da água, o fundo de um rio ou um obstáculo, estas apresentam uma oportunidade para que um sinal emitido seja recebido várias vezes por um recetor. Na figura 3.5 é possível observar um exemplo do cenário descrito. Para um sinal enviado por um emissor, o recetor irá receber o mesmo sinal várias vezes, uma vez que este é refletido pela superfície da água e pelo fundo. No entanto, estes erros na receção podem ser minimizados partindo do pressuposto que o sinal correto será o que irá percorrer o caminho mais curto e, conseqüentemente, irá chegar primeiro ao recetor. Uma vez que percorre o caminho mais curto e que não sofre nenhuma reflexão, que originaria perda de energia por parte do sinal, o sinal correto teoricamente será também o sinal que chega com maior intensidade ao recetor. Porém, uma vez que diferentes superfícies originarão diferentes perdas de energia no sinal refletido, a análise e separação das diferentes receções por parte do recetor torna-se difícil. Dada a característica irregular das superfícies onde o sinal é refletido, é também provável que uma incidência de um sinal numa superfície origine várias reflexões do mesmo sinal, levando à sobreposição de vários sinais acústicos, causando interferência tanto destrutiva como construtiva no sinal recebido [34].

Este fenómeno é especialmente problemático em meios com águas pouco profundas e rochosas, tais como zonas costeiras ou rios, uma vez que não só o meio é muito propício a reflexões, como também existe a possibilidade de não haver um caminho desimpedido entre emissor e recetor, levando a que as únicas captações sejam de sinais refletidos o que, no caso de sistema de posicionamento acústico, origina cálculos de distância incorretos [2].

Tal como referido no subcapítulo 3.1, a velocidade de propagação de sinais acústicos

não é constante num meio subaquático, uma vez que está dependente dos valores de temperatura, pressão e salinidade do meio. Se ocorrer uma mudança substancial de algum destes fatores ao longo da trajetória do sinal entre emissor e recetor, este efetuará uma trajetória curvilínea, tal como pode ser observado no exemplo presente na figura 3.6, em que  $T$ ,  $P$  e  $S$  representam temperatura, pressão e salinidade, respetivamente. A mudança de direção de um sinal acústico como consequência da mudança de velocidade de propagação do sinal é apelidada de refração.

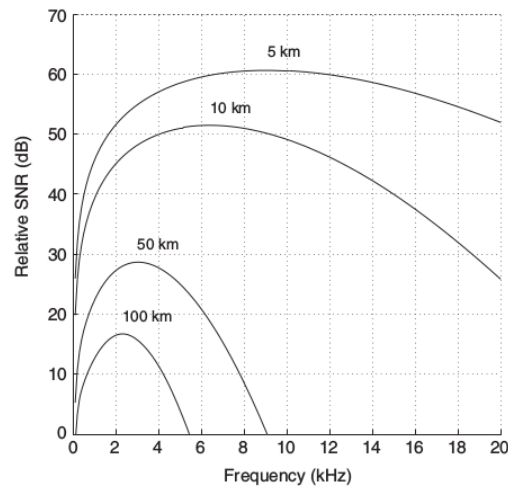


Figura 3.4: Relação entre frequência e SNR do sinal para diferentes alcances [2]

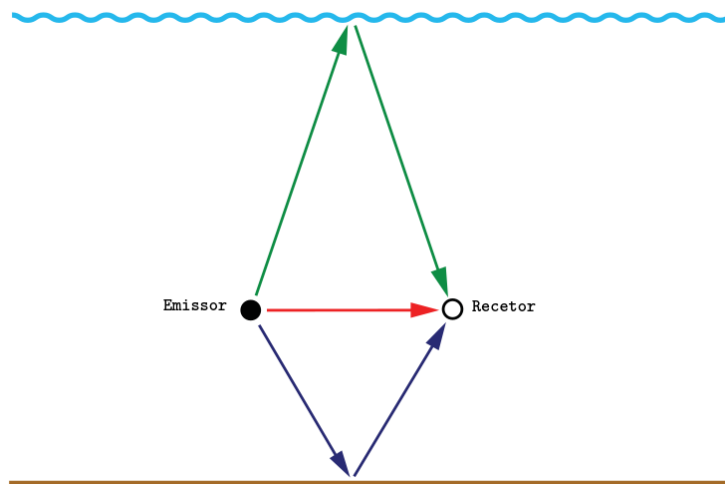


Figura 3.5: Reflexão de um sinal acústico em ambiente subaquático (baseado em [34])

Observando a figura 3.1, é facilmente perceptível que este fenómeno é particularmente prejudicial num cenário que exija uma trajetória com uma grande variação de profun-

didade por parte do sinal enviado. Num exemplo em que a velocidade diminua com a profundidade, verifica-se que a trajetória do sinal curva para baixo ao longo do tempo, de acordo com a lei de Snell [36]. Isto faz com que um sinal que originalmente se dirigia para a superfície possa encaminhar-se para um recetor localizado originalmente à mesma profundidade do emissor (exemplo exposto na figura 3.6). Apesar de fazer uma trajetória maior, uma vez que o faz a uma velocidade superior, pode inclusive chegar ao recetor antes do sinal que percorreu o caminho mais curto e direto. Por outro lado, como percorreu uma maior distância, este sinal chegará com menor intensidade [34].

Perto da superfície, tanto a pressão como a temperatura são relativamente constantes, resultando numa velocidade de propagação constante. Assim sendo, fenómenos de refração são especialmente prejudiciais em aplicações a grandes profundidades e que exijam uma distância considerável de propagação dos sinais utilizados [36].

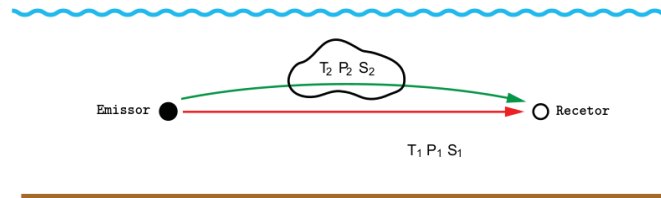


Figura 3.6: Exemplo de refração de um sinal acústico (baseado em[34])

### 3.4 Efeito de Doppler

O movimento do emissor de um sinal em relação ao recetor causa uma variação da frequência do sinal na perspectiva do recetor, fenómeno este conhecido por efeito de Doppler. Para um emissor que se aproxima de um recetor, a frequência do sinal recebido irá ser superior à original, acontecendo o contrário no cenário oposto [37]. A alteração à frequência do sinal imposta pelo efeito de Doppler pode ser descrita através da equação 3.3, em que  $f'$  representa a frequência observada,  $f$  a frequência original,  $v$  a velocidade de propagação do sinal,  $v_o$  a velocidade do recetor e  $v_s$  a velocidade do emissor.

$$f' = \left( \frac{v + v_o}{v - v_s} \right) f \quad (3.3)$$

Uma vez que a velocidade de um sinal acústico em meios subaquáticos é baixa quando comparada com a velocidade de ondas eletromagnéticas, o efeito de Doppler torna-se especialmente prejudicial em sistemas acústicos de comunicação. Os AUV deslocam-se geralmente a velocidades na ordem de poucos metros por segundo mas, apesar disso, os instrumentos presentes em ambientes aquáticos estão sujeitos a correntes, ondas e marés, que podem originar deslocações significativas [36]. As conseqüentes mudanças na frequência

do sinal recebido são prejudiciais, uma vez que dificultam a sua interpretação por parte do recetor. Isto é particularmente notório em sistemas de curto alcance, uma vez que tendencialmente recorrem a sinais de alta frequência [2]. Este fenómeno é também útil, uma vez que permite, por exemplo, estimar a velocidade de alvos analisando a alteração da frequência do sinal captado (este efeito é utilizado, por exemplo, em sensores do tipo *Doppler Velocity Log* (DVL) normalmente utilizados em veículos subaquáticos, uma vez que permitem a estimação da velocidade dos mesmos em relação ao fundo subaquático) [38], [39].

## Capítulo 4

# Formulação do Problema

Tal como referido no capítulo 1, um sistema de posicionamento subaquático enquadra-se nas necessidades imediatas do LSA, uma vez que o laboratório trabalha com vários projetos que envolvem o uso de veículos subaquáticos.

Devido à grande diversidade de missões realizadas pelo laboratório, que por sua vez usam diferentes tipos de veículos, alguns deles veículos autónomos subaquáticos, é importante a existência de um sistema de localização que possa ser utilizado quer a partir da superfície, quer de forma autónoma, entre diferentes sistemas subaquáticos, com um alcance de pelos menos 200 m. Por forma a facilitar a integração e interoperabilidade com sistemas e robôs existentes, o sistema de posicionamento deve ser implementado usando *robot operating system* (ROS), permitindo também a obtenção de dados em tempo real.

Tal como exposto no capítulo 2, um grande número dos sistemas de posicionamento existentes usa medições de TOF para calcular a sua posição em relação a estações em pontos conhecidos. Apesar das medições de TOF resultarem num algoritmo de posicionamento relativamente simples, estas exigem que tanto o alvo como as estações possuam sistemas com capacidade de receção e envio de sinais acústicos, uma vez que a comunicação ocorre nos dois sentidos (o sinal necessita de repetir o trajeto no sentido inverso para que uma medição seja efetuada). Como alternativa, podem ser usadas medições de TDOA que, apesar de resultarem num algoritmo de posicionamento mais complexo e exigirem sincronização temporal entre os diferentes sistemas, permitem um posicionamento através de uma comunicação de apenas um sentido entre emissores e alvo a localizar.

Tendo isto em conta, o sistema de posicionamento acústico aqui exposto recorre à medição de TDOA e posterior cálculo de posição com recurso a um algoritmo de posicionamento hiperbólico.

Um sistema que possua o processamento do lado do alvo tem a vantagem de permitir a coexistência de vários alvos diferentes na mesma missão a usar sistemas semelhantes, sem ser necessária uma adaptação de *hardware* ou *software* existentes nos alvos para que tal aconteça. O sistema envia também um sinal codificado com a sua posição para a superfície

em casos de missões supervisionadas em tempo real e a partir da superfície, para que assim a base da missão possa reconhecer a posição em que o alvo se encontra. Para reduzir a probabilidade de uma má receção da mensagem enviada numa circunstância hipotética em que vários alvos tentam comunicar ao mesmo tempo, a modulação da mensagem enviada é feita com frequências diferentes.

Este sistema de posicionamento requer um conjunto de sistemas de menor dimensão a trabalhar em conjunto, de forma a tornar a localização de um alvo possível. Na superfície, existem três emissores estáticos colocados em posições conhecidas, a transmitir periodicamente um sinal acústico modulado. No alvo a localizar, existe um recetor com a função de capturar os sinais emitidos pelos emissores e, posteriormente, calcular a sua posição com base nos valores de TDOA medidos. Também no alvo, existe um emissor que, periodicamente, irá transmitir um sinal acústico modulado com a informação da sua posição. Por fim, à superfície, existe outro recetor com a função de captar a mensagem e decodificar a posição do alvo.

Para permitir o posicionamento de um alvo a três dimensões usando um método baseado na medição de TDOA, é necessário possuir informação que permita estimar a diferença do alvo para quatro pontos conhecidos. No entanto, este requisito pode ser parcialmente satisfeito, transformando um problema de posicionamento tridimensional numa adaptação de um problema de posicionamento bidimensional. Neste cenário alternativo, apenas é necessária a estimação em relação a três pontos conhecidos, estando a estimação da profundidade a cargo de um sensor de pressão. Este foi o cenário escolhido para este trabalho uma vez que, regra geral, todos os veículos autónomos subaquáticos possuem sensores de pressão para determinar a sua profundidade.

Na figura 4.1 é possível observar um esquema representativo do sistema de posicionamento projetado. As três estações presentes na superfície,  $E1$ ,  $E2$  e  $E3$ , com posições conhecidas  $X_e$ ,  $Y_e$  e  $Z_e$ , conseguem obter através de um sistema GPS não só as suas próprias posições, como também um sinal que permite a sua sincronização temporal (por exemplo, um sinal do tipo *pulse per second* existente nos sistemas GPS). Desta forma, é possível a emissão simultânea do sinal acústico correspondente por parte das três estações. Para facilitar a receção e decodificação dos três sinais acústicos pelo recetor (diminuindo a probabilidade de interferência), o envio dos sinais é desfasado entre os emissores, sendo depois essa diferença compensada na receção do sinal. Assim, após receberem o sinal de sincronismo esperado, as diferentes estações aguardam diferentes instantes de tempo ( $\Delta t$ ) até enviarem o respetivo sinal (em que, no caso da implementação realizada,  $\Delta t1 = 0$  s,  $\Delta t2 = 0.1$  s e  $\Delta t3 = 0.2$  s). A posição  $X_n$ ,  $Y_n$  e  $Z_n$  de um alvo  $A_n$  é estimada recorrendo à fusão de informação proveniente de um sensor de pressão e das medições de TDOA registadas. O TOF de um sinal acústico oriundo da estação 1 e que é captado pelo alvo 1 é representado por  $t_{11}$ , podendo ser transformado numa distância entre a estação e o alvo,  $d_{11}$ , quando a velocidade de propagação no meio é conhecida. Após serem medidas

as diferenças temporais de chegada dos sinais a um alvo, é possível estimar as diferenças de distância do alvo em relação às estações, o que permite o uso de um algoritmo de posicionamento hiperbólico como, por exemplo, o presente em [23], para estimar a posição do alvo.

Periodicamente, após um certo número de leituras, a última posição conhecida pelo alvo é codificada num sinal acústico e enviada para a superfície, para depois ser captada e decodificada pela base de operações da missão. Este método de envio minimiza a probabilidade do sinal enviado com a posição codificada coexistir no meio com os sinais oriundos das estações. No caso de vários alvos partilharem o mesmo espaço de missão, a codificação dos seus valores de posição deverá ser feita com frequências diferentes, possibilitando a sua receção e interpretação simultânea pelo recetor da superfície.

Na figura 4.2 é possível observar uma representação gráfica do envio do sinal para a superfície, onde  $B$  representa a base de operações à superfície.

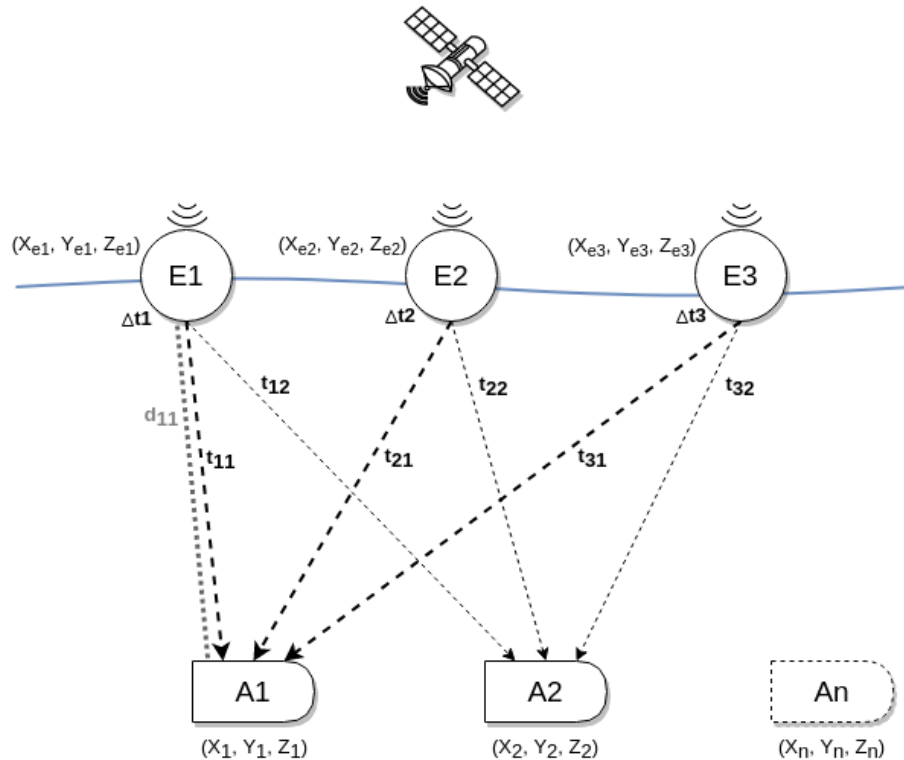


Figura 4.1: Sistema de posicionamento

## 4.1 Arquitetura do Sistema

Uma arquitetura de alto nível do sistema pode ser observada na figura 4.3, onde uma estação emissora, um alvo e um sistema de recepção da base de operações se encontram representados.

Após a recepção de um sinal de *trigger*, o emissor codifica o sinal a enviar, enviando-o de seguida depois do condicionamento apropriado. O recetor condiciona os sinais gravados, efetuando a leitura dos desfasamentos de chegada entre eles. Posteriormente, a sua posição é calculada e codificada. Periodicamente, o sinal codificado é gerado, condicionado e enviado para a superfície. Na superfície, um recetor condiciona o sinal gravado, procedendo à sua descodificação e interpretação da posição enviada pelo alvo.

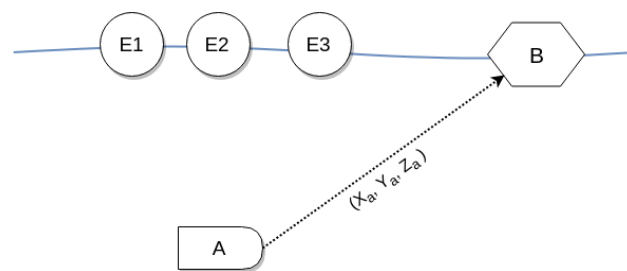


Figura 4.2: Envio da mensagem para a superfície

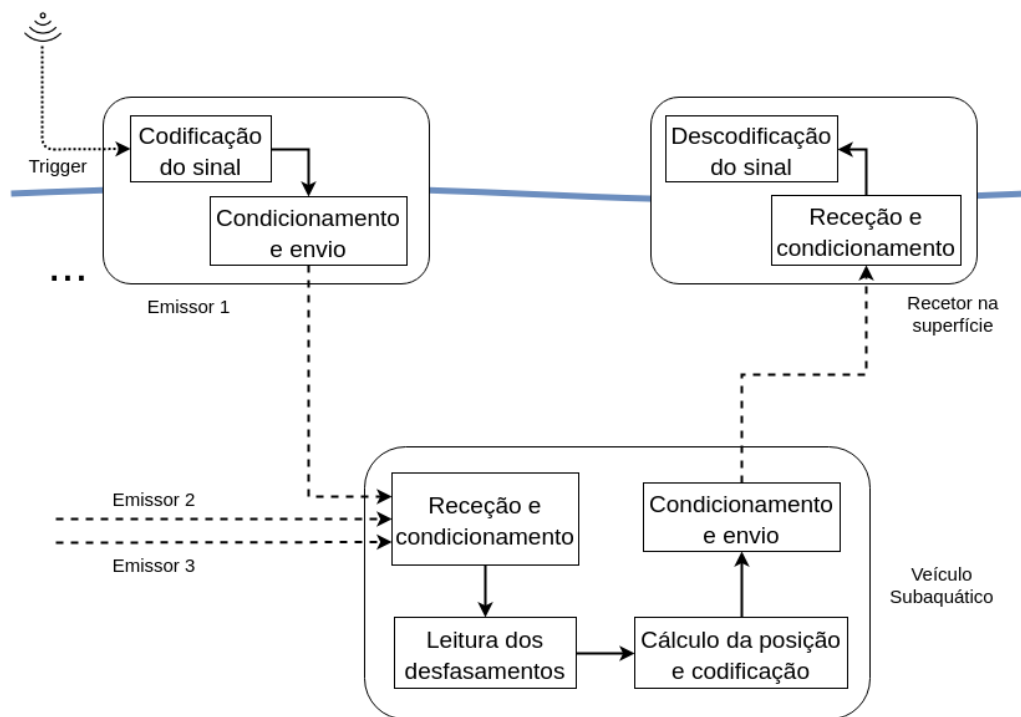


Figura 4.3: Arquitetura do sistema

## Capítulo 5

# Deteção e Codificação Acústica

Numa fase inicial deste capítulo é abordado o trabalho prévio desenvolvido no contexto do projeto MYTAG, trabalho esse que permitiu adquirir os conhecimentos de base necessários para o desenvolvimento deste sistema. Posteriormente, é feita uma exposição do novo algoritmo de deteção, desenvolvido com o intuito de providenciar uma melhoria em relação ao apresentado em [3]. Na parte final do capítulo, são abordados os métodos de codificação e decodificação acústica desenvolvidos.

### 5.1 MYTAG

Seguindo o enquadramento de um projeto pré-existente, denominado de MYTAG, cujo objetivo principal se centrava no estudo do ciclo de vida das solhas-das-pedras, o estudo apresentado em [3] foi realizado. Este projeto pretendeu obter um método que tornasse possível o estudo da localização das solhas de forma pouco intrusiva, uma vez que o método usado previamente para o estudo do seu ciclo de vida consistia na sua captura e identificação com uma etiqueta plástica, estando depois dependente da colaboração de terceiros no momento da sua pesca, para que as suas movimentações pudessem ser registadas.

Utilizando *tags V7* comercializadas pela **Vemco**, em junção com três recetores VRW2, era esperado conseguir criar um sistema que permitisse fazer o posicionamento de forma não intrusiva, após as *tags* terem sido cirurgicamente colocadas em solhas. No entanto, uma vez que os dados devolvidos pelos recetores estavam formatados para o uso num sistema de posicionamento já existente, o *Vemco Positioning System* (VPS), o posicionamento não era possível sem uma posterior troca de dados com a **Vemco**, o que inviabilizou a obtenção da localização das solhas em tempo real [3].

Como tal, foi realizada uma abordagem alternativa ao problema, tendo sido desenvolvido um sistema de receção acústica utilizando como *hardware* uma placa de som UAC-2 da **Zoom** [40] e um hidrofone AS-1, da **Aquarian** [41], permitindo assim a gravação de sinais acústicos com uma taxa de amostragem de 192 kHz.

As gravações com o novo *hardware* ao sinal emitido por uma *tag* V7 permitiram concluir que este era constituído por 8 *pings*, com espaçamentos temporais definidos entre si. Cada *ping* era formado por um sinal do tipo *chirp*, um sinal sinusoidal com frequência variável ao longo do tempo, neste caso particular de aproximadamente 69 kHz. Um exemplo de um sinal emitido por uma *tag* V7 pode ser observado na figura 5.1. Na figura 5.2 é possível observar uma ampliação da porção inicial de um *ping*.

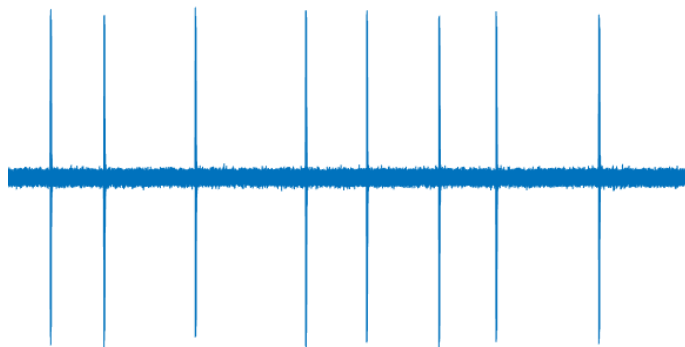


Figura 5.1: Sinal emitido por uma *tag* V7

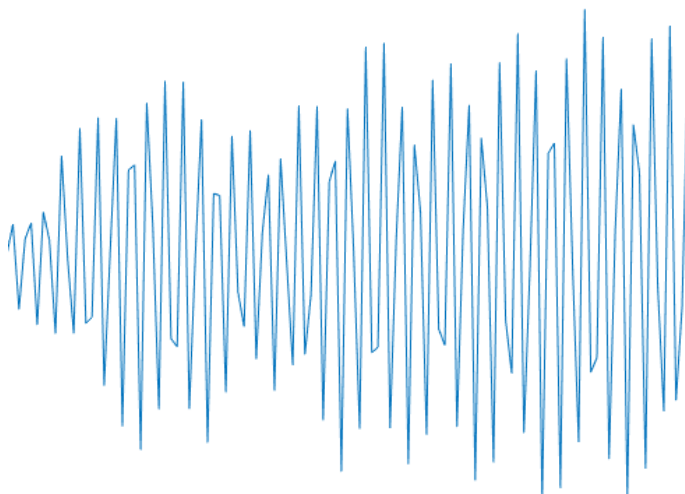


Figura 5.2: Ampliação de um *ping*

Uma análise posterior a sinais enviados por *tags* distintas, permitiu concluir que a codificação do identificador (ID) de cada uma não estava relacionada com os *chirps* em si, mas com o espaçamento temporal entre eles. Concluiu-se, portanto, que cada *tag* teria 7 intervalos temporais únicos entre *pings*, sendo a única exceção o primeiro intervalo, comum a todas as *tags*, representando o *start bit* da mensagem codificada. Isto permitiria

ao recetor distinguir a origem dos diferentes sinais que poderia receber. Para ilustrar esta diferença, na tabela 5.1 estão representados os intervalos temporais de três das *tags* gravadas, em segundos, assim como o seu respetivo ID.

Tabela 5.1: Intervalos entre *pings* nas *tags* V7

ID	$\Delta 1$	$\Delta 2$	$\Delta 3$	$\Delta 4$	$\Delta 5$	$\Delta 6$	$\Delta 7$
1310	0.28	0.48	0.58	0.32	0.38	0.30	0.54
1311	0.28	0.58	0.60	0.58	0.46	0.50	0.30
1312	0.28	0.52	0.58	0.32	0.38	0.38	0.30

Apesar do método de codificação ser conhecido, a relação entre os diferentes intervalos e o número do identificador da *tag* não foi descoberta, uma vez que *tags* com números próximos não apresentavam nenhum padrão aparente. Assim, a única forma de identificar as diferentes *tags* passava por gravar uma vez o sinal emitido por cada uma, adicionando os seus intervalos a uma base de dados. Após a captação de um sinal, os intervalos seriam comparados com os existentes nessa base de dados e, caso fosse encontrada alguma correspondência, a *tag* seria identificada. Tendo em conta tudo o que foi anteriormente referido, foi desenvolvido um algoritmo para identificar os sinais das diferentes *tags* numa gravação de áudio.

Numa fase inicial do algoritmo, o sinal é filtrado com um filtro passa-banda centrado em 69 kHz, sendo o resultado da filtragem correlacionado com um conjunto de amostras representativo de um *ping* esperado. Isto vai originar picos de elevada amplitude nos instantes onde os *pings* se encontram na gravação, sendo a amplitude proporcional à semelhança entre o *ping* esperado e a secção da gravação em questão.

Numa segunda fase, exposta no Algoritmo 1, é feita uma análise ao sinal resultante da correlação, para assim extrair a posição exata dos diferentes *pings*. Primeiramente, é aplicado ao sinal um *threshold* configurável, baseado no valor de *root mean square* (RMS). Todas as amostras do sinal com valores de correlação de amplitude superior ao *threshold* são guardadas num vetor, uma vez que representam possíveis *pings*. Para separar as amostras dos diferentes *pings*, é feita uma análise às posições temporais das amostras que passaram o *threshold*. Amostras em posições consecutivas no vetor que apresentem intervalos temporais significativos (valor de intervalo também configurável) são consideradas como pertencentes a inícios e finais de *pings*. Posteriormente, tendo os limites de cada *ping*, é escolhida a amostra dentro de cada conjunto de limites que apresente maior amplitude de correlação.

Os resultados da aplicação do algoritmo de deteção de *pings* podem ser observados na figura 5.3. A vermelho é possível observar o *threshold* aplicado e também os pontos escolhidos como sendo representativos das posições dos *pings*.

Na figura 5.4 está exposta uma ampliação de um *ping* correlacionado, sendo destacada a vermelho a escolha do algoritmo da amostra representativa da posição de um *ping*.

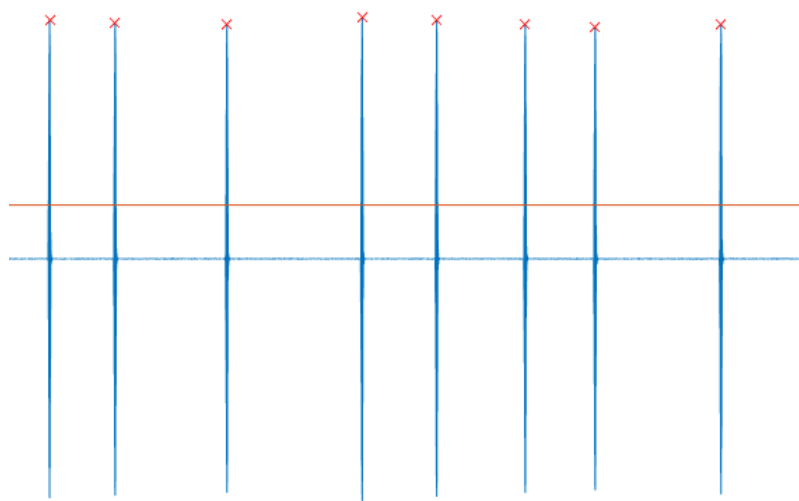


Figura 5.3: Resultados da aplicação do algoritmo de detecção de *pings*

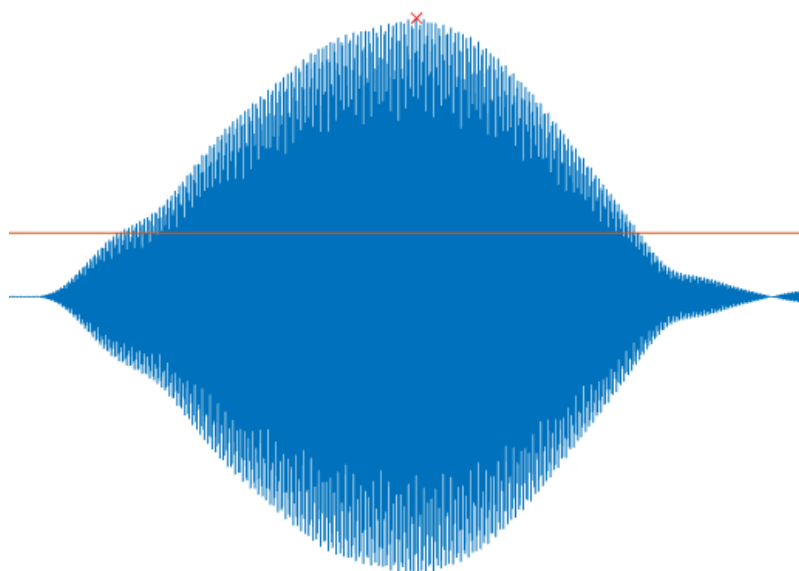


Figura 5.4: Seleção de uma amostra, indicando o instante temporal considerado para um *ping*

---

**Algoritmo 1** Algoritmo de encontrar *pings*

---

**Input:** Amostras de correlação,  $C$ , tamanho do intervalo,  $G$ , e o multiplicador RMS,  $R_M$ .

**Output:** Pings,  $P$ , e intervalo entre pings,  $\Delta P$ .

```
1:  $RMS$ , de  $C$ .
2: for  $i = 0$  até ao tamanho do vetor  $C$  do
3:   if ( $C[i] > RMS * R_M$ ) then
4:     preencher vetor  $Q$  com  $i$ .
5:   end if
6: end for
7: Definir os vetores início de ping,  $P_b$ , e fim de ping,  $P_e$ .
8:  $n_b \leftarrow 1, n_e \leftarrow 0$ .
9:  $P_b \leftarrow Q[0]$ .
10: for  $i = 0$  até ((tamanho do vetor  $Q$ ) - 1) do
11:   if ( $Q[i + 1] - Q[i] > G$ ) then
12:      $P_b[n_b] \leftarrow Q[i + 1]$ .
13:      $P_e[n_e] \leftarrow Q[i]$ .
14:      $n_b \leftarrow n_b + 1, n_e \leftarrow n_e + 1$ .
15:   end if
16: end for
17:  $P_e[n_e] \leftarrow Q[i + 1]$ .
18: Encontrar posições onde os valores de correlação são maiores entre cada  $P_b$  e  $P_e$ .
19: Vetor  $P$  preenchido com a posição do valor máximo encontrado em cada intervalo.
20: Calcular os intervalos entre pings consecutivos,  $\Delta P$ .
21: return  $P$  e  $\Delta P$ .
```

---

Numa fase final, exposta no Algoritmo 2, é realizada uma análise às posições temporais detetadas para os *pings*, para assim tentar descobrir um conjunto de intervalos conhecidos que permitam a identificação de uma *tag*. Esta fase é um pouco mais complexa, uma vez que nem tudo o que a secção anterior considera como possível *ping* efetivamente o é, podendo também aqui existir ruído. Por outro lado, existe também a possibilidade de várias *tags* emitirem num curto intervalo de tempo, levando a que o sinal de várias *tags* diferentes chegue sobreposto ao recetor. Caso duas emitam praticamente em simultâneo, por exemplo, 16 *pings* iriam ser detetados num espaço de gravação que normalmente esperaria ter apenas 8.

Primeiramente, o algoritmo percorre os diferentes intervalos à procura do intervalo correspondente ao *start bit* da mensagem ( $S_\Delta$ ). Caso o intervalo medido entre um determinado possível *ping* e um *ping* mais próximo for inferior ao esperado, o algoritmo calcula sempre a diferença para o *ping* seguinte, e assim sucessivamente, até a diferença encontrada ser a procurada, dentro de uma margem de aceitação configurável, ou exceder o valor esperado, levando a que o *ping* inicial seja descartado como possível início de mensagem e que um novo *ping* seja analisado como possível início. Após ser detetado um início de mensagem, as diferentes *tags* presentes na base de dados são testadas, procurando a sucessão de intervalos que lhes correspondem usando a lógica anterior. Quando uma *tag* é identificada, o seu ID é guardado, assim como a posição temporal de todos os seus *pings*.

---

**Algoritmo 2** Algoritmo para identificar tags

---

**Input:** Pings,  $P$ , intervalos entre pings,  $\Delta P$ , base de dados de tags,  $T$ , e margem,  $M$ .

**Output:** ID das tags,  $T_{ID}$ .

```
1: for  $i = 0$  até ao tamanho do vetor  $\Delta P$  do
2:   for  $h = 0$  até ao tamanho do vetor  $(\Delta P - i)$  do
3:     if  $(h == 0)$  then
4:        $D \leftarrow \Delta P[i]$ .
5:     else
6:        $D \leftarrow \Delta P[i + h] + D$ .
7:     end if
8:     if  $D > S_{\Delta} + M$  then
9:       break.
10:    end if
11:    if  $(D < S_{\Delta} + M$  e  $D > S_{\Delta} - M)$  then
12:       $Matrix[1][j] \leftarrow i, Matrix[2][j] \leftarrow D$ .
13:       $Matrix[3][j] \leftarrow h$ .
14:       $j \leftarrow j + 1$ .
15:      break.
16:    end if
17:  end for
18: end for
19: for  $i = 0$  até ao tamanho de  $Matrix[2]$  do
20:    $found \leftarrow 1, k \leftarrow (Matrix[1][i] + Matrix[3][i])$ .
21:   for  $y = 0$  até ao número de tags guardadas do
22:     for  $j = 1$  to 6 do
23:       for  $h = 0$  até ao tamanho de  $(\Delta P - k)$  do
24:         if  $(h == 0)$  then
25:            $D \leftarrow \Delta P[k + 1]$ .
26:         else
27:            $D \leftarrow \Delta P[k + h + 1] + D$ .
28:         end if
29:         if  $D > (T[j][y] + M)$  then
30:            $found \leftarrow 0$ .
31:           break.
32:         end if
33:         if  $(D < T[j][y] + M$  e  $(D > T[j][y] - M)$  then
34:            $M_D[j - 1] \leftarrow D, k \leftarrow k + h, found \leftarrow 1$ .
35:           if  $(j == 6)$  then
36:             Tag guardada em conjunto com as posições temporais dos pings,  $T_{ID}$ .
37:           end if
38:         else
39:            $found \leftarrow 0$ .
40:         end if
41:       end for
42:     if  $(found == 0)$  then
43:        $k \leftarrow Matrix[1][i] + Matrix[3][i]$ , break.
44:     end if
45:   end for
46:   if  $(found == 1$  e  $j == 6)$  then
47:     break.
48:   end if
49: end for
50: end for
51: return  $T_{ID}$ .
```

---

## 5.2 Detecção de Declive

Apesar de satisfazer os requisitos imediatos do sistema, o algoritmo desenvolvido apresentava alguns problemas, assentes principalmente no uso de um *threshold* calibrável para a separação dos *pings* em relação ao resto do sinal. Apesar deste *threshold* estar dependente do valor de RMS da porção de sinal analisada, em situações que apresentem um SNR variável, ou até em situações em que diferentes sinais com diferentes amplitudes fossem captados simultaneamente, um valor fixo para a calibração deste *threshold* (a variável  $R_M$  no Algoritmo 1) não garantia uma deteção consistente dos vários *pings*. Além disso, este algoritmo era especialmente sensível a fenómenos de reflexão construtiva do sinal (para referência, consultar o subcapítulo 3.3), o que o tornava altamente suscetível a erros em testes realizados no tanque do laboratório. Estes fenómenos por vezes resultam em valores altos de correlação correspondentes a amostras de gravação representativas de sinais refletidos, e não do sinal original que se procura detetar. Assim, uma vez que o Algoritmo 1, após isolar os diferentes *pings*, escolhe como posição temporal de cada *ping* a sua amostra com maior valor de amplitude na correlação, fenómenos de reflexão construtiva inviabilizam uma boa medição dos desfasamentos de chegada dos sinais utilizando este algoritmo. Na figura 5.5 é possível observar um exemplo da influência de fenómenos de reflexão na correlação de um *ping* gravado no tanque do laboratório, onde é perceptível a existência de sinais semelhantes ao inicial, mas captados posteriormente.

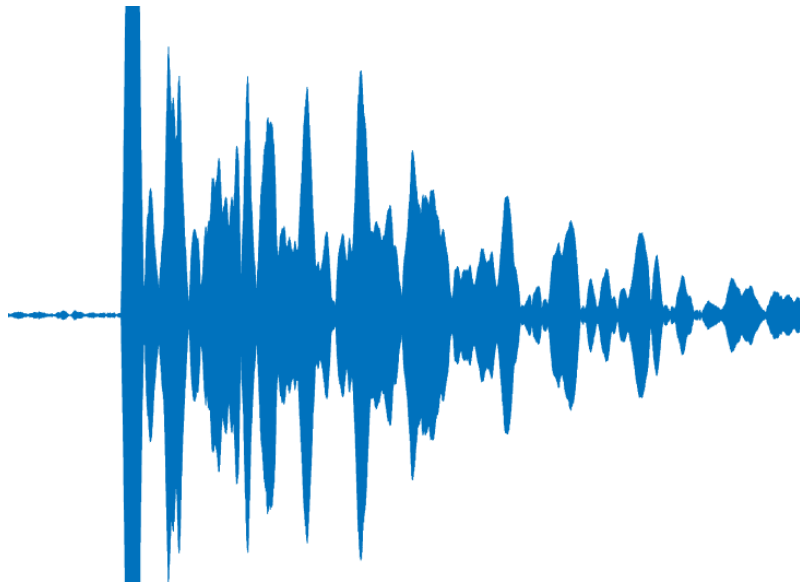


Figura 5.5: Exemplo de reflexão de um *ping*

Para contornar estes problemas, surgiu a necessidade de criar um novo algoritmo de deteção que não se baseasse nos valores mais elevados de amplitude da correlação para identificar um possível *ping*.

Num cenário ideal, com ruído inexistente e sem a ocorrência de outros fenômenos associados à comunicação acústica, o sinal resultante da correlação entre um *ping* gravado e o *ping* esperado deveria apresentar um declive inicial positivo e acentuado, representativo dos instantes em que o sinal gravado começa a apresentar semelhança com o *ping* esperado, um pico no instante de maior semelhança entre os dois e, após esse pico, um declive considerável mas agora negativo, representativo da diminuição da semelhança entre o sinal gravado e o *ping* esperado (um exemplo de uma correlação de um *ping* gravado num cenário próximo da situação ideal está presente na figura 5.4). Por outro lado, quando estamos na presença de uma gravação com reflexões, não só o pico de maior amplitude pode não representar de facto a posição do sinal original, como o declive negativo não será constante, uma vez que os sinais refletidos que chegam mais tarde também apresentam semelhança com o sinal esperado. Considerando esta análise, foi desenvolvido um novo algoritmo, exposto no Algoritmo 3, com foco no único elemento comum a estas duas situações, o declive inicial acentuado, representativo do momento de chegada do sinal que percorreu a menor distância e que, conseqüentemente, não sofreu reflexões.

Uma vez que a correlação é realizada entre dois sinais sinusoidais, o sinal resultante vai ser também ele sinusoidal, centrado a meio da escala. Assim, como o objetivo inicial do novo algoritmo passava por analisar o declive apenas da parte positiva do sinal, os pontos com amplitudes inferiores à amplitude média são descartados.

Após análise a gravações existentes, foi escolhido um valor de declive mínimo,  $m_m$ , entre dois pontos com espaçamento de  $n$  amostras entre si, como sendo representativo da existência de um *ping* no sinal de correlação. Assim, para um ponto na posição  $x$ , de amplitude  $A_x$ , ser considerado como possível início de um *ping*, a condição presente na equação 5.1 teria que ser satisfeita:

$$m_m < \frac{A_{x+n} - A_x}{n} \quad (5.1)$$

A segunda parte do algoritmo começa por isolar num vetor os pontos que satisfazem esta condição. Posteriormente, à semelhança do que acontecia no algoritmo original, os pontos são agrupados caso se verifique um salto temporal grande entre pontos consecutivos nesse vetor, separando assim os pontos pertencentes a *pings* distintos. Por fim são escolhidos, nesses diferentes grupos, os pontos com posição temporal mais baixa, dado que estes representarão o início do *ping*, e calculado o intervalo temporal entre eles. De notar que o importante deste algoritmo não é encontrar com exatidão o ponto inicial do *ping* mas sim encontrar, em todos os *pings*, pontos com posição equivalente, para assim calcular as diferenças de chegada de forma correta.

Na figura 5.6 estão representados os diferentes passos do novo algoritmo de deteção. Na gravação usada como exemplo, estão presentes três *pings*, um deles enviado a uma maior distância do recetor, apresentando conseqüentemente uma amplitude menor. É

possível observar, a azul, os pontos resultantes da correlação que apresentam amplitudes superiores à amplitude média do sinal. A vermelho, os pontos que satisfazem a condição presente na equação 5.1 e, a verde, os três pontos selecionados como representativos da posição de cada *ping*. Na figura 5.7, é possível visualizar com maior detalhe os diferentes pontos selecionados pelo algoritmo, referentes agora apenas a um *ping* e às suas reflexões. De notar, em ambas as figuras, a apropriada escolha do ponto representativo do início de *ping*, escolha essa que é independente das reflexões do sinal e da diferença considerável de amplitudes entre os vários *pings*.

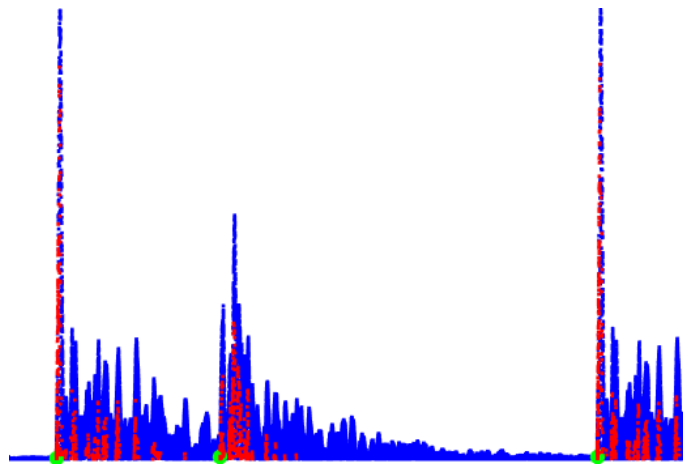


Figura 5.6: Diferentes passos do novo algoritmo de deteção (1)

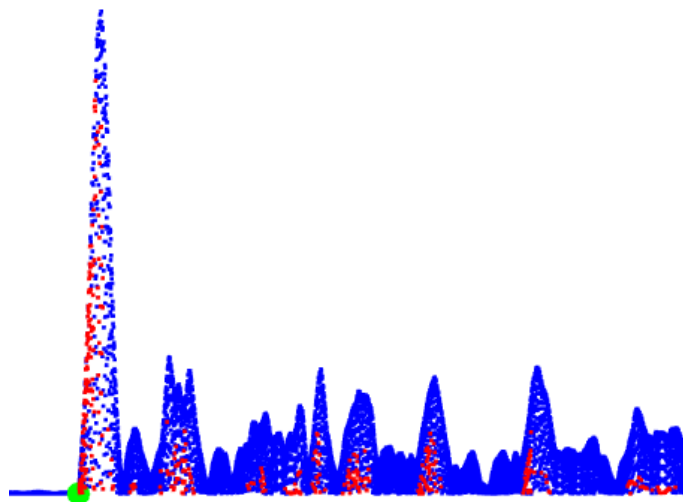


Figura 5.7: Diferentes passos do novo algoritmo de deteção (2)

---

**Algoritmo 3** Novo algoritmo de encontrar *pings*

---

**Input:** Amostras de correlação,  $C$ , tamanho do intervalo,  $G$ , e o declive,  $m_m$ .

**Output:** Pings,  $P$ , e intervalo entre pings,  $\Delta P$ .

```
1: for  $i = 0$  até (tamanho do vetor  $C$ )-1 do
2:   if ( $C[i] > avg(C)$ ) then
3:     preencher vetor  $Q$  com  $i$ .
4:   end if
5: end for
6: for  $i = 0$  até (tamanho do vetor  $Q$ )- $n$  do
7:   if ( $m_m < (Q[i + n] - Q[n])/n$ ) then
8:     preencher vetor  $R$  com  $i$ .
9:   end if
10: end for
11:  $n_b \leftarrow 1$ .
12:  $P \leftarrow R[0]$ .
13: for  $i = 0$  até (tamanho do vetor  $R$  -1) do
14:   if ( $R[i + 1] - R[i] > G$ ) then
15:      $P_b[n_b] \leftarrow R[i + 1]$ .
16:      $n_b \leftarrow n_b + 1$ .
17:   end if
18: end for
19: Calcular os intervalos entre pings consecutivos,  $\Delta P$ .
20: return  $P$  e  $\Delta P$ .
```

---

### 5.3 Funcionalidades de Comunicação

Aproveitando os conhecimentos adquiridos no estudo das *tags* V7 da **Vemco**, uma codificação com lógica semelhante foi criada para os sinais oriundos das três estações.

Para ser usado como *ping*, foi criado um sinal do tipo *chirp* com frequência variável entre 68 kHz e 72 kHz, com a duração de 1 ms. Esta duração baixa foi escolhida não só para minimizar o risco de sobreposição de *pings*, como também para tornar a influência dos sinais refletidos menos proeminente. Uma vez que apenas era necessária a distinção entre três sinais, não era relevante a codificação de muita informação, como nas *tags* V7. Desta forma, para cada sinal foram utilizados apenas três *pings*, em que o primeiro intervalo entre *pings*, à semelhança do que acontecia com as *tags* V7, representa o *start bit*, e o segundo intervalo permite a distinção entre os três sinais.

A escolha do valor dos intervalos foi feita de forma a minimizar a probabilidade de ocorrer a sobreposição de *pings* na recepção, considerando um desfasamento temporal de envio dos três sinais de 0.1 s. Na tabela 5.2 estão presentes os intervalos entre os *pings* dos sinais oriundos de cada estação, em segundos. Estes sinais podem também ser observados na figura 7.19.

Tabela 5.2: Intervalos entre *pings* dos sinais das estações

Estação	$\Delta 1$	$\Delta 2$
A	0.6	0.8
B	0.6	1.0
C	0.6	1.3

Para a decodificação dos sinais, foi implementada uma adaptação do Algoritmo 2. Numa primeira fase, o algoritmo procura pelos três intervalos de *start bit*. Caso os três intervalos sejam encontrados, os intervalos seguintes são medidos, testando não só o intervalo até ao *ping* mais próximo como para os seguintes, tentando assim encontrar os valores esperados referentes às estações.

Na codificação relativa à mensagem a transmitir pelo sistema de emissão presente no alvo, foi utilizada a mesma metodologia. De salientar que com isto não se pretendeu desenvolver estritamente um *modem* acústico, mas sim permitir a transmissão da posição para a superfície utilizando o mesmo tipo de sinais e algoritmos utilizados para se obter posicionamento do sistema.

Neste cenário é necessária não só a codificação acústica da posição do alvo, usando os valores de latitude, longitude e profundidade, como também um valor para servir como ID do alvo, para assim cobrir a possibilidade de implementação do sistema num cenário onde vários alvos partilhem o mesmo espaço de missão em simultâneo. Desta forma, o recetor existente na base de operações pode associar mensagens recebidas a diferentes alvos.

Para diferenciar os *pings* das duas codificações, de forma a que os diferentes sistemas de receção filtrem apenas os *pings* que lhes são respetivos, o *chirp* que serve como *ping* para a codificação da posição apresenta um valor de frequência variável entre 79 kHz e 82 kHz.

Numa fase inicial, foi considerada a hipótese de codificar cada algarismo com um intervalo diferente. Esta hipótese acabou por ser descartada, uma vez que a existência de vários intervalos diferentes aumentava consideravelmente a probabilidade de falsas leituras, dado que um pequeno erro na medição de desfasamentos facilmente implicaria uma interpretação errada da informação.

Para minimizar o número de intervalos utilizados, e assim ter maior margem de manobra na escolha dos mesmos, foi definido que a mensagem seria codificada só após ser colocada em formato binário. Desta forma, apenas com dois intervalos diferentes e um extra para representar o *start bit*, a codificação da informação da posição e do ID torna-se possível. Com o intuito de trabalhar apenas com valores positivos, aos valores medidos de latitude e longitude seriam incrementados  $90^\circ$  e  $180^\circ$ , respetivamente, sendo depois estes valores compensados quando a mensagem fosse captada pelo recetor. Foi também definido que o ID teria que ser um valor entre 0 e 31 e a profundidade máxima medida de cerca de 4000 m, encontrando um equilíbrio entre o número de *bits* utilizado para cada componente e o intervalo de valores que esse número de *bits* permite representar.

Na tabela 5.3 está exposto o número de *bits* usado na representação das componentes decimais e inteiras das diferentes variáveis.

Desta forma, com o envio de 80 *pings*, ou 79 intervalos, torna-se possível codificar a mensagem de posição. Para o *start bit* foi definido um intervalo de tempo de 0.06 s, para o *bit* 0 um intervalo de 0.08 s e para o *bit* 1 um intervalo de 0.04 s. No cenário mais

Tabela 5.3: Número de *bits* atribuídos a cada variável

Variável	Intervalo	n <sup>o</sup> <i>bits</i> parte inteira	n <sup>o</sup> <i>bits</i> parte decimal
ID	0 a 31	5	0
Latitude	0 a 180	8	17
Longitude	0 a 360	9	17
Profundidade	0 a 4095	12	10

demorado, em que a mensagem codificada possui 78 *bits* 0, a duração do sinal é de 6.3 s. Como tempo médio, num sinal que apresente igual número de *bits* 0 e *bits* 1, a duração é de 4.74 s. Um exemplo de uma mensagem gerada pode ser observado na figura 7.21.

Para permitir a interpretação da mensagem codificada, um novo algoritmo de descodificação foi desenvolvido, tendo como base o Algoritmo 2. Este algoritmo, que pode ser observado no Algoritmo 4, permite a descodificação dos valores de ID, latitude, longitude e profundidade a partir dos intervalos de chegada dos diferentes *pings*.

Numa primeira fase do algoritmo, é feita uma procura por um intervalo correspondente a um *start bit*. Caso um seja encontrado, os intervalos seguintes são medidos. De cada vez que haja uma correspondência entre o valor medido e o valor esperado para um dos *bits*, o *bit* correspondente é guardado num vetor. Caso se verifiquem 78 correspondências consecutivas, os *bits* presentes no vetor são separados por variável e posteriormente transformados para o formato original.

---

**Algoritmo 4** Algoritmo de decodificação de mensagem

---

**Input:** Intervalos entre pings,  $\Delta P$ , Intervalos a procurar  $\Delta C$ , e margem,  $M$ .

**Output:** ID, *iden*, latitude, *lat*, longitude, *lon*, e profundidade, *prof*.

```
1: for  $i = 0$  até ao tamanho do vetor  $\Delta P$  do
2:   for  $h = 0$  até ao tamanho do vetor  $(\Delta P - i)$  do
3:     if  $(h == 0)$  then
4:        $D \leftarrow \Delta P[i]$ .
5:     else
6:        $D \leftarrow \Delta P[i + h] + D$ .
7:     end if
8:     if  $D > \Delta C[2] + M$  then
9:       break.
10:    end if
11:    if  $(D < \Delta C[2] + M$  e  $D > \Delta C[2] - M)$  then
12:       $Matrix[j][1] \leftarrow i$ 
13:       $Matrix[j][2] \leftarrow D$ 
14:       $Matrix[j][3] \leftarrow h$ 
15:      break.
16:    end if
17:  end for
18: end for
19: for  $i = 0$  até ao tamanho de Matrix do
20:    $j \leftarrow (Matrix[k][0] + Matrix[k][2] + 1)$ .
21:   while  $j <$  tamanho de  $\Delta P$  do
22:      $h \leftarrow 0$ 
23:     while 1 do
24:       if  $(h == 0)$  then
25:          $D \leftarrow \Delta P[j]$ .
26:       else
27:          $D \leftarrow D + \Delta P[j + h]$ .
28:       end if
29:       if  $(D > \Delta C[0] + M)$  then
30:          $fim \leftarrow 1$ 
31:         break
32:       end if
33:       if  $(D > \Delta C[0] + M)$  e  $(D > \Delta C[0] + M)$  then
34:          $CODE \leftarrow 0$ 
35:         break
36:       end if
37:       if  $(D > \Delta C[1] + M)$  e  $(D > \Delta C[1] + M)$  then
38:          $CODE \leftarrow 1$ 
39:         break
40:       end if
41:        $h \leftarrow h + 1$ 
42:     end while
43:      $j \leftarrow j + h$ 
44:     if  $(fim == 1)$  then
45:       break
46:     end if
47:   end while
48:   if (tamanho de CODE igual a 78) then
49:      $msg \leftarrow CODE$ 
50:   end if
51: end for
52: Separar os bits guardados em msg
53: Transformar os valores binários
54: return iden, lat, lon e prof.
```

---

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

## Capítulo 6

# Posicionamento Hiperbólico

Neste capítulo será abordado, numa primeira fase, o algoritmo hiperbólico utilizado e, posteriormente, a implementação de um filtro de Kalman para auxílio na estimação da posição do alvo.

### 6.1 Algoritmo Hiperbólico

Tal como referido no subcapítulo 2.2, o algoritmo de posicionamento hiperbólico escolhido foi o apresentado por B. T. Fang em [23].

A partir da estimação do TDOA de dois sinais distintos captados por um recetor, oriundos de emissores também eles distintos e com posições conhecidas, é possível posicionar o recetor ao longo de uma hipérbole referenciada a esses emissores. Com a adição de um terceiro sinal proveniente de um terceiro emissor, torna-se exequível posicionar o recetor na interseção de um conjunto de hipérboles. Para estimar o ponto desta interseção onde o recetor se encontra, é necessária informação adicional como, por exemplo, a profundidade a que o alvo se encontra em relação ao referencial definido pelos emissores. A computação de uma posição com um método hiperbólico pode ser reduzida à resolução de equações quadráticas, usando as *baselines* definidas pelos diferentes emissores como referência e explorando as propriedades geométricas da interseção de hipérboles [23].

Um exemplo de uma interseção hiperbólica a duas dimensões pode ser observado na figura 6.1, onde a partir dos valores das diferenças de chegada do sinal oriundo de três emissores, representados a vermelho, é possível delinear três hipérboles, estando o recetor localizado num dos dois pontos (assinalado na imagem) em que estas se intersejam.

Apesar da resolução apresentada por B. T. Fang ser relativa a um cenário onde o alvo a estimar possui um emissor, e os recetores estão colocados em posições conhecidas, a mesma lógica de raciocínio pode ser aplicada a um cenário inverso, onde o recetor está no alvo que se pretende localizar e os diferentes emissores se encontram em posições conhecidas.

Na figura 6.2 é possível observar a posição de navegação relativamente a três estações.

A estação  $A$ , na posição  $(0, 0, 0)$ , define a origem do referencial, a estação  $B$ , na posição  $(b, 0, 0)$ , situa-se num ponto pertencente ao eixo  $XX$ , enquanto que a estação  $C$ , na posição  $(c_x, c_y, 0)$ , localiza-se num ponto arbitrário pertencente ao plano  $XY$ . Sendo  $V$  a velocidade de propagação do sinal,  $T_a, T_b$  e  $T_c$  os tempos de chegada do sinal às diferentes estações,  $R_{ab}$  a diferença entre a distância do alvo à estação  $A$  e a distância do alvo à estação  $B$ , e  $R_{ac}$  a diferença entre a distância do alvo à estação  $A$  e a distância do alvo à estação  $C$ , é possível obter as equações 6.1 e 6.2.

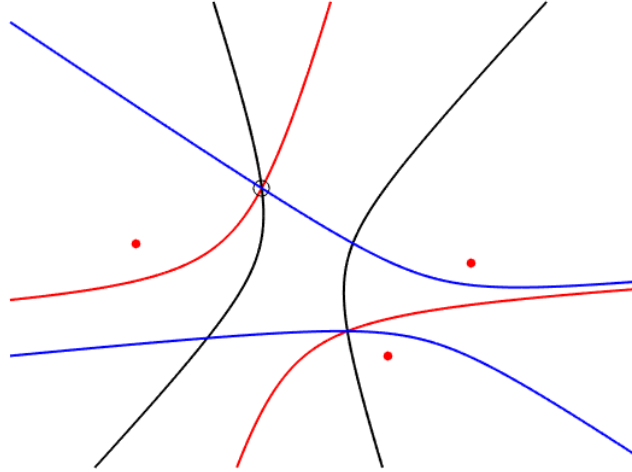


Figura 6.1: Interseção hiperbólica

$$R_{ab} = V \cdot (T_a - T_b) = \sqrt{x^2 + y^2 + z^2} - \sqrt{(x - b)^2 + y^2 + z^2} \quad (6.1)$$

$$R_{ac} = V \cdot (T_a - T_c) = \sqrt{x^2 + y^2 + z^2} - \sqrt{(x - c_x)^2 + (y - c_y)^2 + z^2} \quad (6.2)$$

Após simplificação, e assumindo que  $c = \sqrt{c_x^2 + c_y^2}$ , resultam as equações 6.3 e 6.4.

$$R_{ab}^2 - b^2 + 2 \cdot b \cdot x = 2 \cdot R_{ab} \cdot \sqrt{x^2 + y^2 + z^2} \quad (6.3)$$

$$R_{ab}^2 - c^2 + 2 \cdot c_x \cdot x + 2 \cdot c_y \cdot y = 2 \cdot R_{ac} \cdot \sqrt{x^2 + y^2 + z^2} \quad (6.4)$$

Partindo destas equações, e caso  $R_{ab}$  seja diferente de zero, é possível obter a equação 6.5, em que  $g$  pode ser calculado através da equação 6.6 e  $h$  pela equação 6.7.

$$y = g \cdot x + h \quad (6.5)$$

$$g = \{R_{ac} \cdot (b/R_{ac}) - c_x\} / c_y \quad (6.6)$$

$$h = \{c^2 - R_{ac}^2 + R_{ac} \cdot R_{ab}(1 - (b/R_{ab})^2)\} / 2 \cdot c_y \quad (6.7)$$

Substituindo a equação 6.5 na equação 6.3, resulta a equação 6.8, em que  $d$ ,  $e$  e  $f$  podem ser calculados usando as equações 6.9, 6.10 e 6.11, respectivamente.

$$z = \pm \sqrt{d \cdot x^2 + e \cdot x + f} \quad (6.8)$$

$$d = - \{1 - (b/R_{ab})^2 + g^2\} \quad (6.9)$$

$$e = b \cdot \{1 - (b/R_{ab})^2\} - 2 \cdot g \cdot h \quad (6.10)$$

$$f = (R_{ab}^2/4) \cdot \{1 - (b/R_{ab})^2\}^2 - h^2 \quad (6.11)$$

Caso o valor de altitude ou profundidade  $z$  seja conhecido, então a posição do alvo pode ser calculada através da resolução da equação 6.12 [23].

$$d \cdot x^2 + e \cdot x + (f - z^2) = 0 \quad (6.12)$$

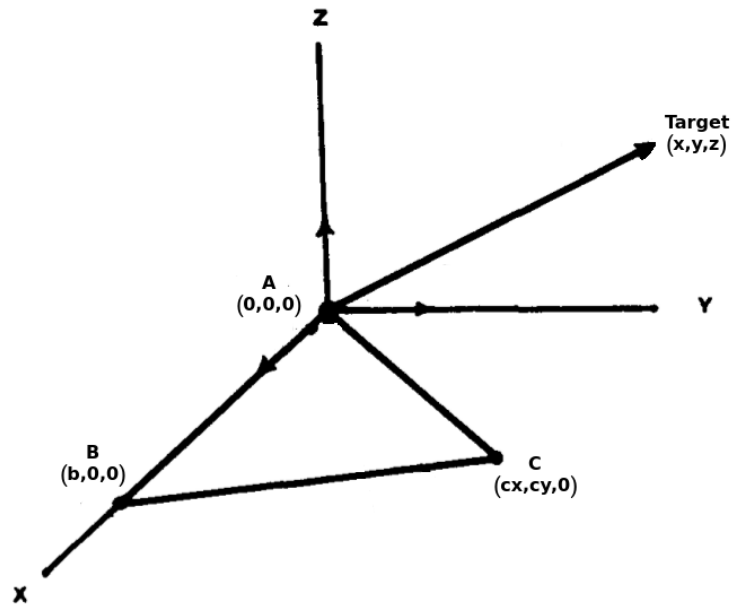


Figura 6.2: Posição relativa ao referencial definido pelos recetores (baseado em [23])

Tal como referido anteriormente, para a utilização do algoritmo exposto por B. T. Fang, a posição das estações teria que satisfazer determinadas condições. Estas necessitam de gerar um referencial em que a origem é definida pela posição da estação  $A$ , e o eixo  $XX$  definido pela posição da estação  $B$ . No entanto, num cenário real isto não é viável, uma vez que liberdade na escolha das posições das diferentes estações é algo necessário. Para que esta liberdade na escolha de posições esteja presente e o algoritmo possa ser implementado, as posições das diferentes estações são alteradas virtualmente. Após ser aplicado o algoritmo, ao resultado é aplicada uma transformação inversa à realizada anteriormente, permitindo a obtenção da posição real do alvo a partir da posição calculada.

Para simplicidade da explicação, os valores das posições são aqui representados no formato  $P(x, y, z)$ . Para alterar as posições das estações de forma a que a posição da estação  $A$  seja  $A(0, 0, 0)$ , é aplicada uma translação às posições das estações de valor igual ao inverso da posição atual de  $A$ , tal como ilustrado na equação 6.13, em que  $P_X$ ,  $P_Y$  e  $P_Z$  representam os valores da posição da estação após a translação,  $X_a$ ,  $Y_a$  e  $Z_a$  representam os valores da posição da estação  $A$  antes de sofrer a translação e  $P_x$ ,  $P_y$  e  $P_z$ , representam os valores da posição da estação antes de sofrer a translação.

$$\begin{bmatrix} P_X \\ P_Y \\ P_Z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -X_a \\ 0 & 1 & 0 & -Y_a \\ 0 & 0 & 1 & -Z_a \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \quad (6.13)$$

A estação  $A$  estará agora na origem do referencial, faltando agora aplicar uma rotação no eixo  $ZZ$  à posição das estações, para colocar a estação  $B$  na posição  $B(0, y, z)$ . A todos os pontos é posteriormente aplicada uma rotação em  $ZZ$  com um valor de ângulo inverso ao valor do ângulo formado entre a posição atual da estação  $B$  e o eixo  $XX$  (obtido através da equação 6.14, em que  $Y$  e  $X$  representam os valores da posição da estação  $B$  em  $x$  e  $y$ , respetivamente). Este cálculo está exemplificado na equação 6.15, onde  $\theta$  representa o ângulo referido.

$$\theta = \text{atan}_2(Y/X) \quad (6.14)$$

$$\begin{bmatrix} P_X \\ P_Y \\ P_Z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \quad (6.15)$$

Após o algoritmo ser aplicado, são realizadas translações e rotações inversas às aplicadas anteriormente, para assim colocar os pontos devolvidos pelo algoritmo no seu referencial original.

Uma vez que a interseção das três hipérbolas origina duas soluções diferentes e, consequentemente, a equação final apresentada por B. T. Fang origina também ela duas raízes distintas, foi desenvolvido um algoritmo para resolver este problema. Neste algoritmo, com base nas duas possíveis localizações para o alvo, os desfasamentos esperados são calculados e comparados com os que foram realmente obtidos, escolhendo posteriormente a posição que apresente menor erro.

Numa primeira instância, é verificado qual o sinal que chegou primeiro ao recetor, representando assim a estação mais próxima. Posto isto, os tempos de chegada são referenciados em relação ao primeiro sinal a ser captado. Como exemplo, a equação presente em 6.16 representa o cenário em que o primeiro sinal a ser captado foi o oriundo da estação  $A$ , em que  $t_a$ ,  $t_b$  e  $t_c$  representam o instante de captação dos sinais provenientes das estações  $A$ ,  $B$  e  $C$ , respetivamente.

$$u0 = [t_b - t_a, t_c - t_a] \quad (6.16)$$

Posto isto, é calculado o equivalente à equação 6.16 para as duas posições devolvidas pelo algoritmo. Para tal, primeiramente são calculadas as distâncias entre cada posição e as três estações. Como exemplo, a equação 6.17 representa o cálculo da distância entre a primeira posição devolvida pelo algoritmo e a base  $A$ , em que  $X_a$ ,  $Y_a$  e  $Z_a$  representam os valores da posição da estação, e  $x_1$ ,  $y_1$  e  $z_1$  os valores da possível posição do alvo.

$$d_{a1} = \sqrt{(x_1 - X_a)^2 + (y_1 - Y_a)^2 + (z_1 - Z_a)^2} \quad (6.17)$$

Após a distância às três estações estar calculada, é verificada qual a estação mais próxima e criado um vetor equivalente ao apresentado em 6.17. Como exemplo, a equação presente em 6.18 demonstra um cenário em que a possível posição do alvo a ser testada encontra-se mais próxima da base  $A$ , em que  $v$  representa a velocidade de propagação no meio,  $d_{a1}$ ,  $d_{b1}$  e  $d_{c1}$  as distâncias entre a posição e as estações  $A$ ,  $B$  e  $C$ , respetivamente.

$$u1 = [0, (d_{b1} - d_{a1})/v, (d_{c1} - d_{a1})/v] \quad (6.18)$$

Numa fase final, é calculado o erro entre  $u0$  e os vetores criados para as duas possíveis posições, escolhendo a posição que apresente erro menor. Este cálculo está presente na equação 6.19. Como exemplo, é aqui apresentado o cálculo do erro entre o vetor  $v0$  e o vetor gerado para a primeira posição,  $v1$ .

$$e_1 = \sum_{i=0}^2 |u0_i - u1_i| \quad (6.19)$$

Numa implementação prática deste método de posicionamento, um sistema de coordenadas como o aqui referido pode não ser obtido de forma direta. No caso de serem

usados sensores de GPS nos emissores, as suas posições geográficas serão conhecidas a partir dos seus valores de latitude, longitude e profundidade, o que não permite relacionar as diferentes posições segundo a lógica exposta na figura 6.2.

No entanto, este problema pode ser contornado achando os valores equivalentes para um sistema de coordenadas diferente como, por exemplo, o *Universal Transverse Mercator* (UTM). Num sistema de coordenadas UTM, a superfície terrestre é dividida numa grelha, sendo a posição geográfica de um ponto descrita pela identificação da célula da grelha em que se encontra, e pela sua posição bidimensional, em metros, dentro dessa célula [42].

Desta forma, uma vez que o alvo a posicionar conhece as posições dos emissores, estas posições podem ser convertidas para um sistema de coordenadas como o UTM, possibilitando a criação do sistema de coordenadas exposto na figura 6.2. Após o algoritmo de posicionamento ser aplicado, a conversão inversa pode ser efetuada, possibilitando ao alvo o reconhecimento dos valores de latitude, longitude e profundidade da sua posição atual. Posto isto, estão reunidas as condições para que a posição possa ser codificada e enviada acusticamente com o método exposto no subcapítulo 5.3.

## 6.2 Filtro de Kalman

Uma vez que as medições de desfasamento estão sujeitas a erros, podendo até não se realizar em determinadas circunstâncias, e que o modelo da dinâmica do alvo a seguir era linear, foi implementado um filtro de Kalman linear para estimar a posição do alvo ao longo do tempo.

Apresentado por R. E. Kalman em [43], um filtro de Kalman é um algoritmo que utiliza medições efetuadas ao longo do tempo, com ruído associado, para produzir uma estimação dos seus valores reais, permitindo assim uma estimação ponderada do estado de um sistema dinâmico num determinado momento.

Este algoritmo tem duas fases: uma fase de previsão, em que o filtro produz uma estimação das variáveis de estado do instante atual e das suas incertezas a partir da estimação do instante anterior, e uma fase de correção, em que informação proveniente de observações é usada para atualizar a estimação da previsão, com maior peso a ser dado às estimações que possuam menor incerteza.

Um filtro de Kalman assume que as variáveis de estado são aleatórias e gaussianas, com valor médio  $\mu$ , que representa o centro da distribuição, e com variância  $\sigma^2$ , que representa a incerteza das mesmas. Além disso, é assumido também que a dinâmica pode ser descrita por sistemas lineares, o que permite que as relações entre os diferentes estados possam ser impostas por equações diferenciais lineares.

Na fase inicial do algoritmo, a fase de previsão, o estado atual,  $\hat{x}_k$ , é calculado em função do estado anterior,  $\hat{x}_{k-1}$ , através da equação 6.20, em que  $A_k$  representa a matriz de previsão, que relaciona o estado anterior com o atual, e  $B_k$  e  $u_k$ , a matriz e o vetor

de controlo, respetivamente, que representam influências externas ao sistema. Nesta fase é também calculada a nova matriz de covariância, ou de incerteza,  $P_k$ , em função da covariância do instante anterior,  $P_{k-1}$ , através da equação 6.21, em que  $Q_k$  representa o ruído associado ao modelo do sistema.

$$\hat{x}_k = A_k \cdot \hat{x}_{k-1} + B_k \cdot u_k \quad (6.20)$$

$$P_k = A_k \cdot P_{k-1} \cdot A_k^T + Q_k \quad (6.21)$$

A fase de correção acontece de cada vez que há observações, servindo para atualizar as estimações de estado e de covariância tendo como base medições de sensores. Uma vez que as observações podem não traduzir diretamente o estado que se pretende estimar, é necessária a criação de uma matriz  $H_k$ , que relaciona as observações com o estado a estimar. Para pesar a estimacão resultante da fase de previsão com as observações, definindo assim a confiança depositada em cada um dos passos, é calculado o ganho de Kalman,  $K_k$ , através da equação 6.22, em que  $R_k$  representa o ruído associado às medições dos sensores.

$$K_k = P_k \cdot H_k^T \cdot (H_k \cdot P_k \cdot H_k^T + R_k)^{-1} \quad (6.22)$$

Tendo o ganho de Kalman, é possível calcular a nova estimacão do estado,  $\hat{x}_{k|k}$ , através das medições realizadas pelos sensores,  $z_k$ , com a equação 6.23. Por fim, a nova matriz de covariância,  $\hat{P}_{k|k}$ , pode ser obtida utilizando a equação 6.24, em que  $I$  representa a matriz identidade, ficando assim concluído o algoritmo do filtro de Kalman.

$$\hat{x}_{k|k} = \hat{x}_k + K_k \cdot (z_k - H \hat{x}_k) \quad (6.23)$$

$$P_{k|k} = (I - K_k H) \cdot P_k \quad (6.24)$$

Na próxima iteração do filtro, as matrizes de estado e de covariância calculadas servirão como inicialização das equações da fase de previsão, de forma cíclica.

Para avaliar o desempenho do filtro (tendo em vista a afinação do mesmo), dois dos indicadores mais significativos são a inováção, presente na equação 6.25 e a covariância da inováção, presente na equação 6.26.

$$\tilde{y}_k = z_k - H \cdot \hat{x}_k \quad (6.25)$$

$$S_k = H_k \cdot P_k \cdot H_k^T + R_k \quad (6.26)$$

Um dos testes comuns para testar a consistência de um filtro de Kalman, passa por verificar a consistência dos valores de inovação,  $\tilde{y}_k$ , de cada estado em relação à sua covariância,  $S_k$ . Tal pode ser verificado caso cerca de 95 % dos valores de inovação estejam compreendidos entre  $\pm 2\sqrt{S_k}$ . Muitas vezes este teste é suficiente para verificar a consistência do filtro, no entanto é boa prática aplicar também os testes de seguida apresentados [44].

Para confirmar que o filtro não apresenta nenhuma tendência, é calculada a inovação normalizada,  $q_k$ , através da fórmula presente na equação 6.27, para um sequência de  $i$  iterações do filtro.

$$q_k(i) = \tilde{y}_k(i) \cdot S_k^{-1} \cdot \tilde{y}_k(i) \quad (6.27)$$

Tendo os valores da inovação normalizada, a média deslizando,  $\bar{q}_k$ , pode ser calculada com a expressão presente na equação 6.28, em que  $N$  representa o tamanho da janela usada.

$$\bar{q}_k(i) = \frac{1}{N} \cdot \sum_{i=1}^N q_k(i) \quad (6.28)$$

Para validar que o filtro não apresenta nenhuma tendência, o valor final da média deslizando terá que estar compreendido entre os valores ( $r_1$  e  $r_2$ ) apresentados na equação 6.29. Como exemplo, a equação 6.29 apresenta os valores de fronteira resultantes para uma confiança de 95 % e um valor de  $N$  de 100.

$$\begin{aligned} [r_1, r_2] &= [\chi_{100}^2(0.025), \chi_{100}^2(0.975)] \\ &= [74.22, 129.6] \end{aligned} \quad (6.29)$$

Por fim, para testar o ruído branco presente na inovação, é realizada uma autocorrelação para os valores de inovação, de acordo com a equação 6.30, sendo depois a autocorrelação normalizada por  $r(0)$ . A autocorrelação terá um pico em  $\tau = 0$  e, idealmente, será aleatoriamente distribuída à volta de 0 nos pontos seguintes, tendo por isso que apresentar uma média muito próxima de 0 para que o teste seja bem sucedido [44].

$$r(\tau) = \frac{1}{N} \cdot \sum_{k=0}^{N-\tau-1} \tilde{y}_k^T \cdot \tilde{y}_k + \tau \quad (6.30)$$

Tendo em conta o movimento tridimensional de um alvo genérico, a sua posição num determinado instante ( $x_k, y_k$  e  $z_k$ ) pode ser calculada, em função da sua posição anterior ( $x_{k-1}, y_{k-1}$  e  $z_{k-1}$ ), usando a equação 6.31, em que  $\Delta t$  representa o intervalo temporal

entre iterações e  $v_x, v_y$  e  $v_z$  as velocidades no eixo  $x, y$  e  $z$ , respetivamente.

$$\begin{cases} x_k = x_{k-1} + v_x \cdot \Delta t \\ y_k = y_{k-1} + v_y \cdot \Delta t \\ z_k = z_{k-1} + v_z \cdot \Delta t \end{cases} \quad (6.31)$$

Dado que as medições sensoriais apenas permitem calcular a posição do alvo, o vetor de estados usado no filtro possui os valores de posição e velocidade:

$$\hat{x} = [x \ y \ z \ v_x \ v_y \ v_z] \quad (6.32)$$

Utilizando a equação 6.31, é possível definir a matriz de previsão  $A$ , presente na equação 6.33, responsável por relacionar o estado da iteração anterior com o estado atual.

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.33)$$

A matriz  $Q$ , que representa o ruído associado ao modelo do sistema, e que entra no cálculo da covariância na fase de previsão, está exposta na equação 6.34, onde  $\sigma$  representa o desvio padrão do erro associado a cada um dos estados.

$$Q = \begin{bmatrix} \sigma_x & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{v_z} \end{bmatrix} \quad (6.34)$$

O vetor de observações,  $z$ , resultante da aplicação do algoritmo hiperbólico aos desfazamentos medidos, e a matriz  $H$ , que relaciona este mesmo vetor com o estado a estimar, podem ser observados nas equações 6.35 e 6.36, respetivamente.

$$z = [x \ y \ z] \quad (6.35)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (6.36)$$

A matriz  $R$ , que representa o ruído associado às observações, está representada na equação 6.37, onde  $\sigma_m$  representa o desvio padrão do erro associado a cada uma das medições.

$$R = \begin{bmatrix} \sigma_{m_x} & 0 & 0 \\ 0 & \sigma_{m_y} & 0 \\ 0 & 0 & \sigma_{m_z} \end{bmatrix} \quad (6.37)$$

Com o intuito de facilitar a convergência do filtro, este só é iniciado a partir do momento em que é registrada uma segunda leitura de desfasamentos e conseqüente cálculo de posição. Assim, os valores iniciais de posição do filtro correspondem à posição obtida a partir da primeira medição, sendo os valores de velocidade iniciados a zero. A matriz de covariância inicial é também ela inicializada a zeros.

## Capítulo 7

# Emissão e Recepção

Uma vez que o *hardware* utilizado previamente em [3] não satisfazia os requisitos agora existentes, foi desenvolvido um novo sistema de recepção, com uma maior capacidade de aquisição e com uma melhor amplificação do sinal recebido, assim como um emissor capaz de emitir sinais personalizados a distâncias satisfatórias, no contexto do sistema de posicionamento a implementar.

Neste capítulo é exposto o projeto do *hardware* utilizado para permitir o envio e a recepção dos diferentes sinais acústicos, bem como o *software* desenvolvido e implementado no sistema de emissão. O desenvolvimento do *software* do sistema de recepção, uma vez que era comum a outros projetos em desenvolvimento no LSA, foi elaborado por elementos desses projetos, não fazendo por isso parte do trabalho aqui descrito, apesar de ter sido utilizado no sistema final. Além disso, são também apresentados testes feitos aos sistemas elaborados e, por fim, é abordada a implementação em ROS dos algoritmos associados aos sistemas.

### 7.1 Hardware do Sistema de Emissão

O sistema de emissão de sinais desenvolvido tem que ser capaz de, na versão implementada nas diferentes estações, enviar um sinal acústico pré-definido de cada vez que tal for requisitado e, na versão implementada no alvo, codificar uma mensagem recebida por porta série num sinal acústico, enviando-o posteriormente. Para tal, foi utilizado um microcontrolador para a criação do sinal acústico e desenvolvido um módulo para condicionamento de sinal de forma a otimizar ao máximo o sinal a ser transmitido pelo hidrofone utilizado, o AS-1 da **Aquarian** [41].

Foi escolhida a gama de alto desempenho STM32F4, da **STMicroelectronics**, que possui um arquitetura *Advanced RISC Machine* (ARM) e frequências de trabalho até 180 MHz. Por conveniência de acessibilidade ao material, foram utilizadas placas de desenvolvimento NUCLEO-F446RE, que proporcionam um interface físico acessível ao microcon-

trolador nelas presente, neste caso um STM32F446. Este microcontrolador, da gama de alto desempenho STM32F4, possui também características que o tornam compatível com os requisitos do sistema de emissão, tais como 2 *digital-to-analog converter* (DAC) de 12 *bits*, 17 temporizadores e 16 linhas de interrupções externas [45].

No entanto, uma vez que o microcontrolador usado trabalha com uma gama de tensões entre 0 V e 3.3 V, o sinal gerado não estava otimizado em função daquilo que o hidrofone AS-1 está pronto a receber, uma vez que este aceita tensões até 30 V<sub>pp</sub>. Um maior aproveitamento desta gama corresponde a um maior alcance do sinal emitido pelo hidrofone [41]. Desta forma, foi desenvolvida uma *printed circuit board* (PCB) para servir como módulo de condicionamento para o sinal oriundo do DAC, aproveitando assim as características do hidrofone usado. O circuito tem como objetivo não só a amplificação do sinal, como também a subtração ao sinal de uma componente contínua, uma vez que o sinal saído do DAC estava centrado a meio da gama de tensões utilizada pelo microcontrolador. Após esta subtração, o sinal passa a estar centrado em 0 V, aproveitando assim a gama de trabalho disponibilizada pelo hidrofone AS-1.

Desta forma, foi utilizado um amplificador operacional numa montagem subtratora, representada na figura 7.1.

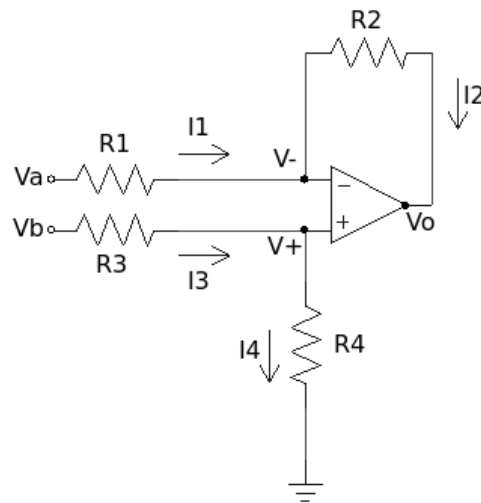


Figura 7.1: Montagem subtratora

Considerando as correntes que passam pelas resistências  $R_3$  e  $R_4$ , é possível concluir que:

$$I_3 = \frac{(V_b - V_+)}{R_3}, \quad I_4 = \frac{V_+}{R_4} \quad (7.1)$$

Uma vez que o amplificador operacional tem um grande valor de impedância na entrada, as correntes  $I_3$  e  $I_4$  vão ser aproximadamente iguais. Desta forma, igualando as expressões das correntes e isolando  $V_+$ , obtém-se:

$$V_+ = V_b \cdot \frac{R_4}{(R_3 + R_4)} \quad (7.2)$$

Seguindo o raciocínio anterior, considerando as correntes que passam nas resistências  $R_1$  e  $R_2$ , é possível deduzir que:

$$I_1 = \frac{(V_a - V_-)}{R_1}, \quad I_2 = \frac{(V_- - V_o)}{R_2} \quad (7.3)$$

Uma vez que  $I_1$  e  $I_2$  são aproximadamente iguais, e isolando  $V_-$ , é possível concluir que:

$$V_- = V_b \cdot \frac{(V_a \cdot R_2 + V_o \cdot R_1)}{(R_1 + R_2)} \quad (7.4)$$

Dado que, num amplificador operacional com realimentação negativa,  $V_+ \approx V_-$ , é possível igualar as expressões presentes em 7.2 e 7.4. Isolando posteriormente  $V_o$ , obtém-se:

$$V_o = V_b \cdot R_4 \cdot \frac{(R_1 + R_2)}{(R_1 \cdot (R_3 + R_4))} \quad (7.5)$$

Considerando um cenário em que  $R_1 = R_3$  e  $R_2 = R_4$ ,  $V_o$  pode ser calculado em função de  $V_a$  e  $V_b$  por [46]:

$$V_o = \frac{R_2}{R_1} \cdot (V_b - V_a) \quad (7.6)$$

Uma vez que o sinal sinusoidal oriundo do DAC estava centrado em cerca de 1.65 V (metade da gama de tensões utilizada pelo microcontrolador), para que esta componente contínua lhe fosse retirada, levando a que este passe a ficar centrado em 0 V, uma tensão de 1.65 V tem que ser aplicada em  $V_a$ . Foi também definido que o sinal pós condicionamento devia ter um valor de cerca de 17 V<sub>pp</sub>, o que representaria um ganho de cerca de 5.6 vezes em relação ao sinal original, que teria cerca de 3.3 V<sub>pp</sub>. Considerando que, no circuito representado na figura 7.1,  $R_1 = R_3 = 10 \text{ k}\Omega$  e  $R_2 = R_4 = 56 \text{ k}\Omega$ , o ganho pretendido é obtido. Recorrendo ao *software Psim*, o circuito aqui exposto foi simulado. Na figura 7.2 está apresentada a relação entre o sinal de entrada, a azul, representativo do sinal gerado pelo DAC, e o sinal de saída, a vermelho, representativo do sinal após o condicionamento, com valores de tensão (V) indicados no eixo vertical e valores temporais (s) no eixo horizontal. Para um sinal de entrada com frequência de 69 kHz, valor médio de 1.65 V e 3.3 V<sub>pp</sub>, é obtido na saída do circuito um sinal com a mesma frequência, mas

com valor médio de 0 V e 17 V<sub>pp</sub>, tal como pretendido.

O amplificador operacional escolhido foi o OPA2192, da **Texas Instruments**. Este amplificador de alta precisão cobre os requisitos do circuito, uma vez que não só trabalha com tensões elevadas (até 36 V), como também apresenta *rail-to-rail input e output*, o que permite a entrada e saída de sinais com tensões até às suas tensões de alimentação, que têm como valor máximo  $\pm 20$  V [47].

Por questões de ordem prática, o módulo de condicionamento de sinal foi desenvolvido com o intuito de encaixar na NUCLEO-F446RE, que ficava encarregue de fornecer a alimentação para o módulo. Deste modo, a alimentação disponível era uma tensão contínua de 5 V, o que não permitia o funcionamento desejado do circuito de condicionamento, uma vez que o amplificador operacional apenas suporta tensões até aos valores das suas alimentações. Para contornar este obstáculo foi utilizado um MAX680, um conversor de tensão do tipo *charge-pump* da **Maxim Integrated**, seguindo a montagem apresentada na figura 7.3. Com este conversor, a partir de uma tensão de 5 V, ficariam também disponíveis para o circuito tensões de 10 V e -10 V, permitindo assim a alimentação positiva e negativa do amplificador operacional [48].

Na figura 7.4, é possível observar o emissor desenvolvido, com o módulo de condicionamento de sinal, a NUCLEO-F446RE e o hidrofone AS-1.

Na figura A.2, presente no apêndice A, está exposto o esquema elétrico do módulo de condicionamento de sinal desenvolvido para o sistema emissor.

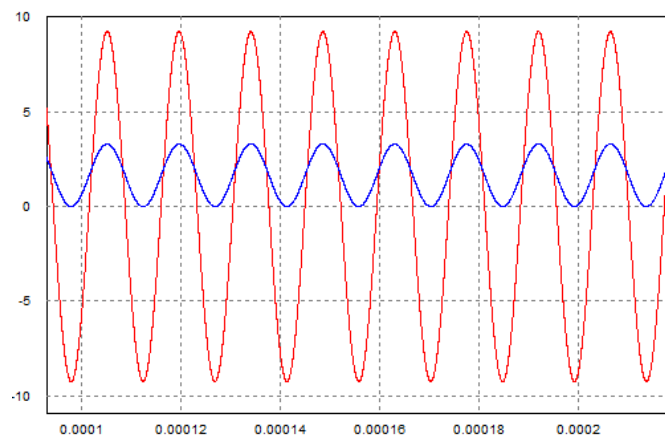


Figura 7.2: Relação entre sinal de entrada e sinal de saída da montagem subtratora

## 7.2 Hardware do Sistema de Recepção

O sistema de recepção desenvolvido tem como objetivo primário servir como melhoria face ao utilizado em [3], tendo que para isso possuir uma taxa de aquisição superior aos 192 kHz disponibilizados pela placa UAC-2 [40], mantendo uma adequada capacidade de

amplificação do sinal recebido. Para tal, foi desenvolvido um módulo para condicionar o sinal oriundo do hidrofone AS-1 e utilizado um microcontrolador que permitia uma taxa alta de aquisição de dados e posterior envio dos mesmos via *ethernet*. Para maximizar a qualidade das gravações, foi também utilizado um filtro passa-alto EF125 da **Thorlabs**, de 50 kHz, para assim filtrar o sinal oriundo do hidrofone [49].

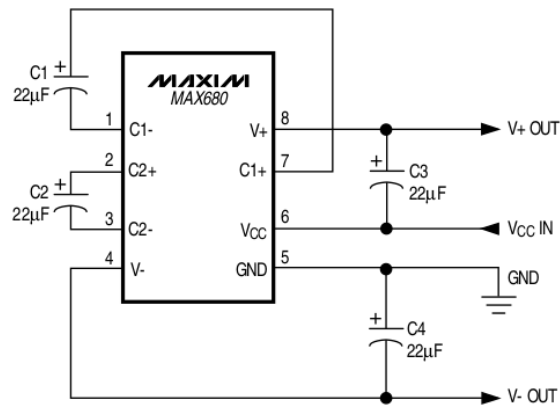


Figura 7.3: Montagem do MAX680 [48]

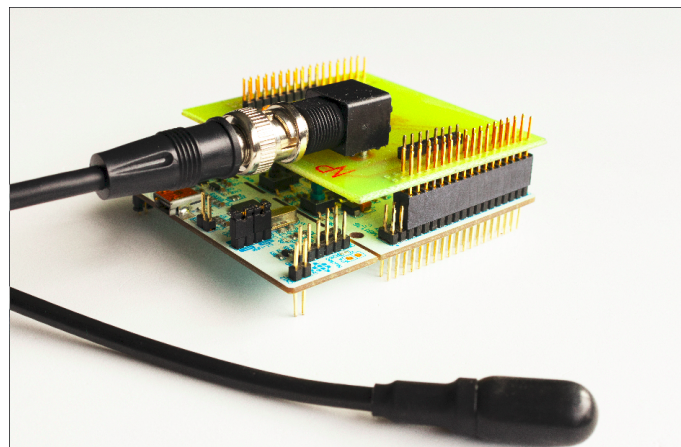


Figura 7.4: Sistema emissor

Tal como no sistema de emissão, para o sistema de receção foi utilizada uma placa de desenvolvimento da **STMicoelectronics**, neste caso uma NUCLEO-H743ZI [50]. Esta placa facilita a interação com o microcontrolador nela presente, um STM32H743ZI que, com uma arquitetura ARM e frequência de trabalho até 400 MHz, se encontra enquadrado na gama de alto desempenho da família STM32. Este microcontrolador satisfaz os requisitos do sistema, uma vez que além de permitir transferência de dados por *ethernet*,

possui também 3 ADC de 16 *bits* e 22 temporizadores [51].

O sinal oriundo do hidrofone era um sinal centrado em 0 V, expectavelmente de baixa amplitude mas que, em casos de sinais sonoros de grande intensidade, poderia atingir valores até  $\pm 4$  V. Uma vez que a gama de tensões aceites pelo microcontrolador se situa entre 0 V e 3.3 V, um módulo de condicionamento de sinal foi desenvolvido não só para otimizar o sinal recebido em função da gama de aceitação do microcontrolador, amplificando-o e adicionando-lhe uma componente contínua, centrando-o a meio dessa gama, como também para proteger o microcontrolador de sinais eléctricos de intensidade elevada que o pudessem danificar. Outro aspeto importante do módulo desenvolvido foi a minimização do ruído acrescentado ao sinal recebido, uma vez que isso influencia diretamente os resultados de qualquer análise posterior que fosse feita ao sinal.

Para condicionamento do sinal foi usado um circuito com 3 andares de amplificadores operacionais, estando o primeiro deles exposto na figura 7.5, uma montagem amplificadora não inversora.

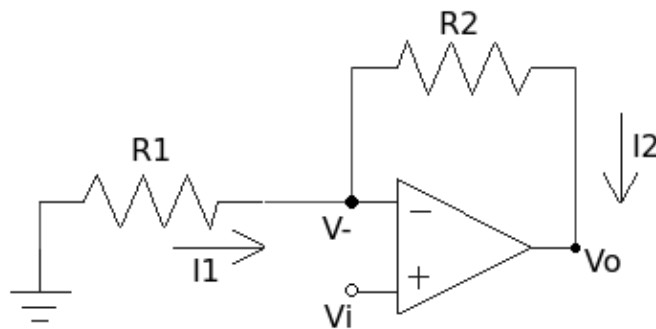


Figura 7.5: Montagem amplificadora não inversora

Uma vez que o amplificador operacional possui entradas com alta impedância, é possível concluir que:

$$V_o = R_1 \cdot I_1 + R_2 \cdot I_2, \quad I_1 = I_2 = \frac{V_-}{R_1} \quad (7.7)$$

Simplificando as expressões, obtém-se:

$$V_o = V_- \cdot \frac{R_1 + R_2}{R_1} \quad (7.8)$$

Considerando que  $V_- \approx V_i$ , e simplificando, conclui-se que  $V_o$  pode ser calculado em

função de  $V_i$  numa montagem amplificadora não inversora através da expressão [46]:

$$V_o = V_i \cdot \left(1 + \frac{R_2}{R_1}\right) \quad (7.9)$$

Após alguns testes no tanque do laboratório com um sistema emissor, foi definido um ganho de aproximadamente 70 vezes como valor ideal, uma vez que para uma distância de poucos metros, o sinal recebido tinha amplitudes de cerca de 0.02 V, permitindo assim amplificar o sinal de forma a este aproveitar a gama de aceitação por parte do microcontrolador, atingindo assim valores de pico de cerca de 1.4 V. Isto deixaria ainda alguma margem para que não acontecesse perda de informação em situações que o emissor estivesse ainda mais próximo e, conseqüentemente, originasse a captação de sinais com maior amplitude. Considerando que, no circuito apresentado na figura 7.5,  $R_1 = 10 \text{ k}\Omega$  e  $R_2 = 680 \text{ k}\Omega$ , o ganho pretendido é obtido. Na figura 7.6 é possível observar os resultados obtidos em simulação para a tensão de saída (a vermelho) em função da tensão de entrada (a azul), um sinal sinusoidal com frequência de 69 kHz e amplitude máxima de 0.02 V, com valores de tensão (V) indicados no eixo vertical e valores temporais (s) no eixo horizontal. O sinal de saída é uma onda com a mesma frequência mas com valores de amplitude máximos de aproximadamente 1.4 V.

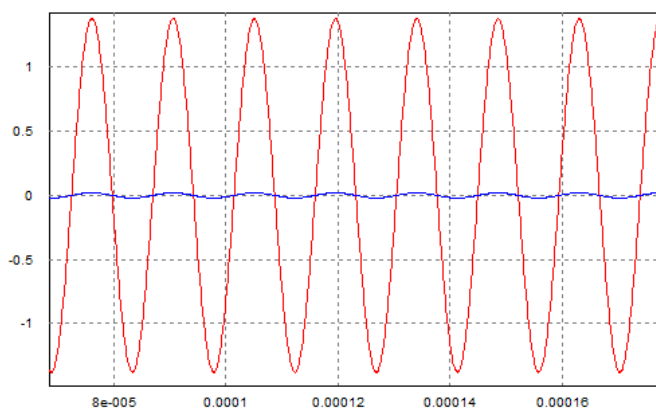


Figura 7.6: Sinal de entrada e de saída da montagem amplificadora não inversora

Para o segundo andar do circuito foi utilizada um montagem somadora inversora, presente na figura 7.7.

Devido às propriedades dos amplificadores operacionais, é possível concluir que:

$$I_3 = I_1 + I_2, \quad V_- \approx V_+ = 0 \quad (7.10)$$

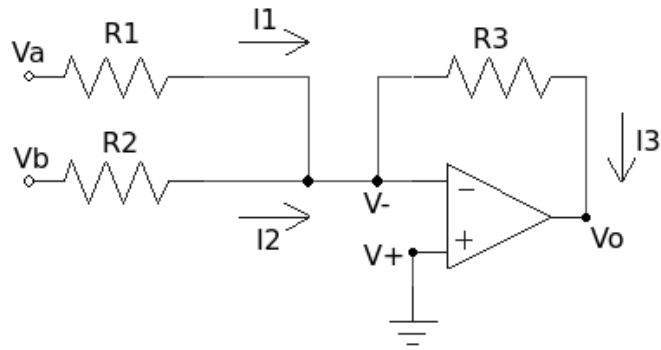


Figura 7.7: Montagem somadora

Posto isto, é viável deduzir que:

$$\frac{-V_o}{R_3} = \frac{V_a}{R_1} + \frac{V_b}{R_2} \quad (7.11)$$

Isolando  $V_o$ , é possível estabelecer a relação entre as tensões de entrada e a tensão de saída através da expressão:

$$V_o = -\left(\frac{R_3}{R_1} \cdot V_a + \frac{R_3}{R_2} \cdot V_b\right) \quad (7.12)$$

No caso particular de  $R_1 = R_2 = R_3$ , a expressão anterior fica simplesmente:

$$V_o = -(V_a + V_b) \quad (7.13)$$

Uma vez que o sinal precisa de ser centrado a meio da gama de aceitação de tensão do microcontrolador, uma tensão de cerca de 1.6 V foi aplicada em  $V_b$ , sendo o sinal oriundo do primeiro andar do circuito aplicado em  $V_a$ . Uma vez que as resistências usadas possuíam o mesmo valor, isto originava um sinal de saída igual ao sinal de entrada, mas com o acrescento de uma componente contínua de -1.6 V. Na figura 7.8 é possível observar os resultados obtidos em simulação para a tensão de saída (vermelho) em função da tensão de entrada (azul), representativa do sinal de saída do primeiro andar, com valores de tensão (V) indicados no eixo vertical e valores temporais (s) no eixo horizontal. O sinal de saída é igual ao sinal de entrada, mas centrado em -1.6 V.

Por fim, no último andar do circuito, foi utilizada uma montagem inversora para assim passar o sinal para dentro da gama de aceitação de tensões do microcontrolador, centrando-o em 1.6 V. Um esquema da montagem inversora pode ser observado na figura 7.9.

Tendo em conta a alta impedância das entradas do amplificador operacional, é possível deduzir que:

$$I_1 = \frac{V_i}{R_1} - \frac{V_-}{R_1} = I_2 = \frac{V_-}{R_2} - \frac{V_o}{R_2} \quad (7.14)$$

Considerando que, devido às propriedades dos amplificadores operacionais,  $V_+ = 0 \text{ V}$  e  $V_- \approx V_+$ , e simplificando, conclui-se que  $V_o$  pode ser calculado em função de  $V_i$  numa montagem amplificadora inversora através da expressão [46]:

$$V_o = V_i \cdot \left(-\frac{R_2}{R_1}\right) \quad (7.15)$$

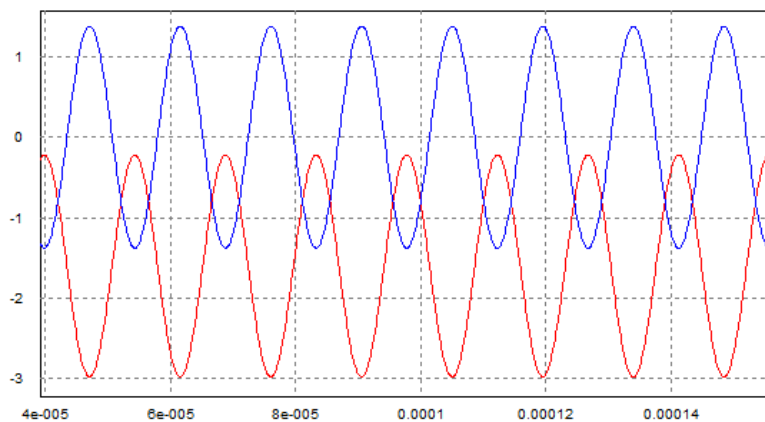


Figura 7.8: Sinal de entrada e de saída da montagem somadora inversora

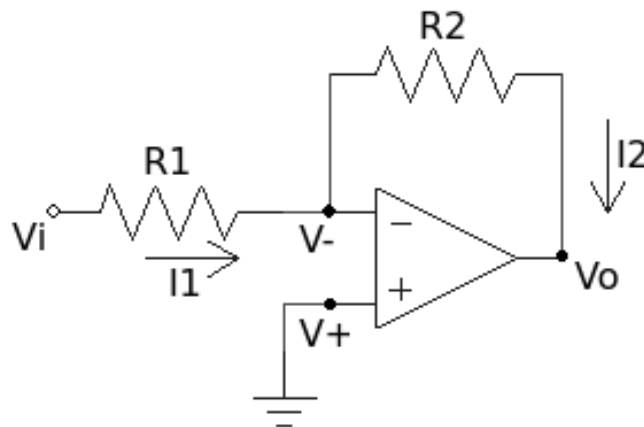


Figura 7.9: Montagem inversora

Uma vez que o desejado deste último andar era apenas a inversão do sinal, e não a imposição de qualquer ganho, os valores de  $R_1$  e  $R_2$  seriam iguais. Na figura 7.10 é possível

observar os resultados obtidos em simulação para a tensão de saída (vermelho) em função da tensão de entrada (azul), representativa do sinal de saída do segundo andar, com valores de tensão (V) indicados no eixo vertical e valores temporais (s) no eixo horizontal. O sinal de saída é o inverso do sinal de entrada, ficando assim enquadrado entre 0 V e 3.3 V, tal como exigido pelo microcontrolador utilizado.

Na figura 7.11 está representado o circuito final de condicionamento de sinal utilizado. Tal como no sistema de emissão, o modelo de amplificadores utilizado foi o OPA2192, da **Texas Instruments** [47]. De notar que o mesmo resultado teria sido conseguido apenas com os últimos dois andares, introduzindo ganho na montagem inversora. No entanto, por conveniência no desenho do módulo de condicionamento de sinal de forma a este poder ser constantemente alterado na fase de testes para diferentes valores de ganho, esta foi a abordagem escolhida.

Para proteção do microcontrolador no caso do sinal captado pelo hidrofone ser de grande amplitude, foi acrescentado na saída do circuito um diodo de Zener de 3.3 V.

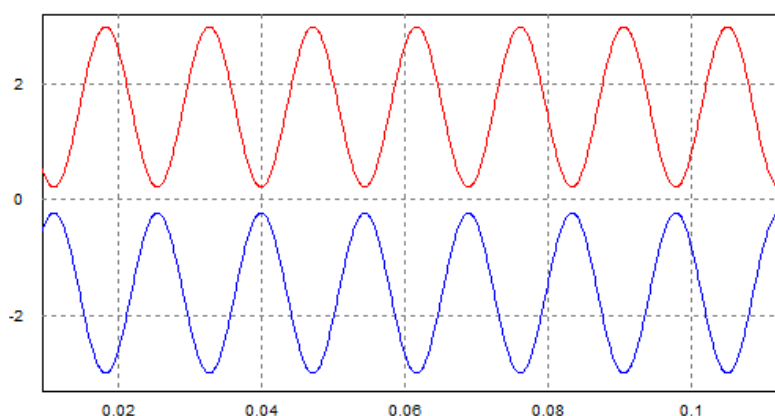


Figura 7.10: Sinal de entrada e de saída da montagem inversora

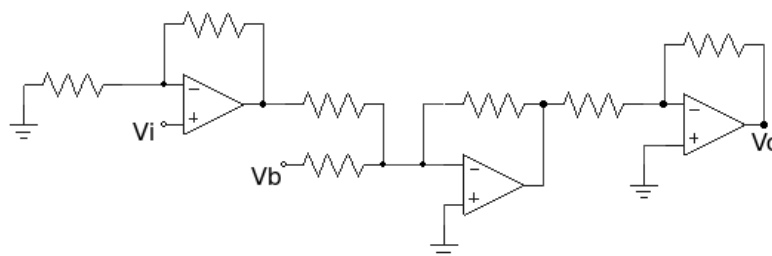


Figura 7.11: Circuito de condicionamento de sinal

Tal como foi definido para o sistema de emissão, também o módulo de condicionamento de sinal do sistema de receção foi desenvolvido com o intuito de encaixar na NUCLEO-

H743ZI. Tendo isto em conta, as alimentações disponíveis não eram suficientes para as exigências do circuito. Após alguns testes, foi também verificado que o conversor do tipo *charge-pump* MAX680 utilizado no sistema de emissão introduzia uma quantidade de ruído considerável ao sinal caso este estivesse responsável pela alimentação dos amplificadores. Tendo isto em conta, foi utilizado um conversor *charge-pump* LM27762 da **Texas Instruments**. O LM27762 caracteriza-se por ser um conversor de baixo ruído, que possibilita a conversão de uma tensão de entrada de 5 V para uma tensão de saída de  $\pm 5$  V, possuindo ainda um *low-dropout regulator* (LDO) na saída negativa, para reforçar a estabilidade do sinal [52]. A montagem realizada e incorporada no módulo de condicionamento de sinal pode ser observada na figura 7.12 e o seu esquema elétrico na figura A.1, presente no apêndice A.

Na entrada do conversor LM27762 foi colocado um filtro passa-baixo do tipo LC, com o intuito de estabilizar o sinal de alimentação oriundo da placa de desenvolvimento NUCLEO-H743ZI. Uma vez que uma boa aquisição de dados por parte deste sistema está muito dependente da quantidade de ruído existente no sinal, foram escolhidos condensadores e bobinas com valores de capacitância e reatância elevados, para assim reduzir ao máximo a frequência de corte aplicada ao sinal por parte do filtro. A relação entre o valor da frequência de corte e os valores de capacitância e reatância no circuito utilizado pode ser descrita através da equação 7.16, em que  $f_c$  representa a frequência de corte,  $L$  o valor de reatância da bobina, e  $C$  o valor da capacitância do condensador [53].

$$f_c = \frac{1}{2 \cdot \pi \cdot \sqrt{L \cdot C}} \quad (7.16)$$

Na figura A.3, presente no apêndice A, é possível observar o esquema elétrico do módulo de condicionamento de sinal do sistema recetor.

Na figura 7.13, é possível observar o sistema de receção desenvolvido, com o módulo de condicionamento de sinal, a NUCLEO-H743ZI, o hidrofone AS-1 e o filtro EF125.

### 7.3 Software do Sistema de Emissão

O *hardware* desenvolvido é comum a dois sistemas de emissão diferentes: o presente nas três estações, que ao receber um sinal de *trigger* externo, envia um sinal codificado com a identificação da respetiva estação, e o presente no alvo, que tem como função codificar a posição atual do mesmo e enviar periodicamente para a superfície o sinal codificado. Numa primeira fase, é aqui exposto o *software* implementado nos sistemas de emissão das estações, sendo posteriormente exposto o presente no sistema de envio de mensagem.

Para gerar o sinal de *trigger* para os diferentes sistemas das estações, simulando assim a presença de um sensor GPS, foi utilizada uma placa de desenvolvimento NUCLEO-F446RE.

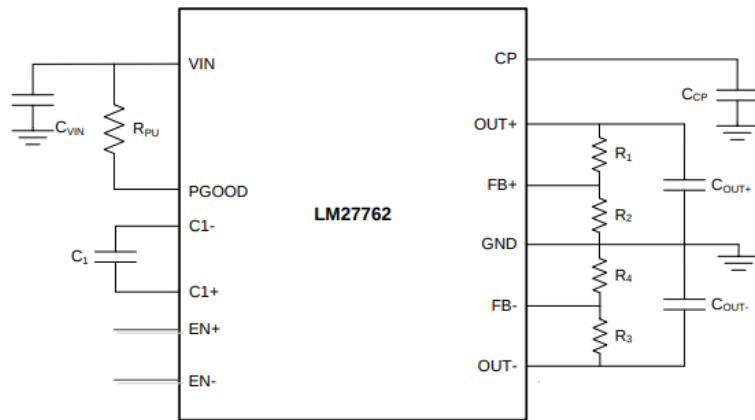


Figura 7.12: LM27762 [52]

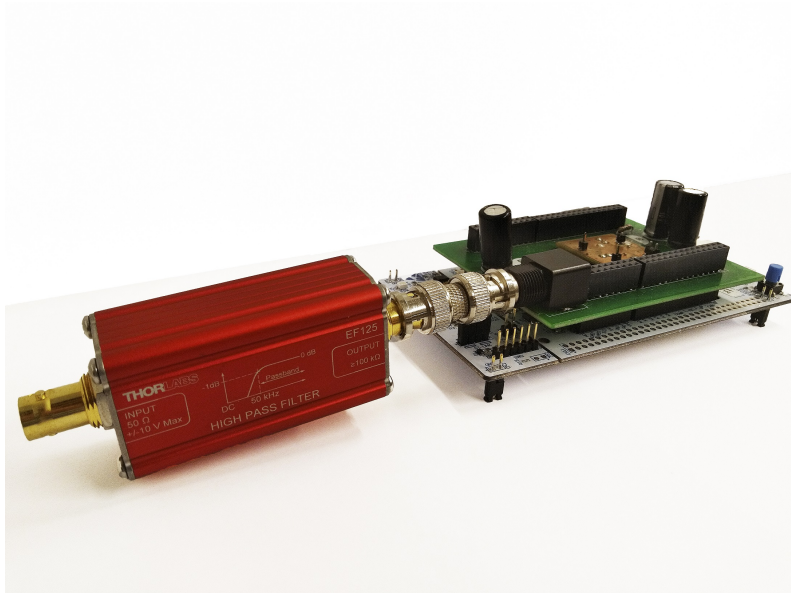


Figura 7.13: Sistema recetor

Para que o posicionamento hiperbólico fosse possível, as três estações teriam que emitir o respetivo sinal em simultâneo, para que depois o desfasamentos de chegada pudessem ser medidos e estimada a posição do alvo. No entanto, o algoritmo de deteção tornava-se consideravelmente mais falível quando vários sinais se sobrepunham quando captados pelo recetor. Para limitar este problema, foi determinado que o sinais iriam ser enviados pelas estações com um desfasamento temporal fixo entre si, sendo depois este desfasamento compensado no algoritmo de receção, levando à obtenção de medidas de desfasamento de chegada equivalentes ao cenário em que as estações enviam em simultâneo os respetivos sinais.

A placa tem assim que gerar periodicamente três sinais de *trigger* com um desfasamento fixo entre si. Para tal, foi configurado um *timer* com um período de 10  $\mu$ s e a sua respetiva interrupção. Desta forma, a porção de código presente na interrupção seria executada com uma periodicidade de 10  $\mu$ s. Dentro da interrupção, uma variável contadora é incrementada, sendo o estado das saídas referentes a cada um dos sinais de *trigger* dependente do valor dessa variável. Após um certo número de incrementos, correspondentes ao período pretendido para a ocorrência de sinais de *trigger*, a variável contadora é reiniciada, voltando o ciclo ao ponto inicial.

Na figura 7.14 é possível observar um diagrama que exemplifica o algoritmo implementado na placa de *trigger*. Tal como é perceptível, alterando os valores de  $T1$ ,  $T2$  e  $T3$ , é possível configurar os diferentes desfasamentos entre os sinais de *trigger*. De notar que a escolha de um desfasamento demasiado elevado se traduz num aumento do erro associado ao posicionamento, pois o alvo receberá os três sinais em posições distintas, devendo por isso a configuração do desfasamento ter isso em atenção. Além disso, configurando o valor de  $T0$ , é alterada a periodicidade deste ciclo e, conseqüentemente, do envio dos sinais de *trigger*. Os sinais resultantes serão ondas quadradas cujo o tempo em nível lógico *high* está dependente dos desfasamentos entre os *triggers* e que apresentarão periodicidade dependente de  $T0$ .

O valor da periodicidade do envio dos sinais de *trigger* pode ser definido alterando o valor de  $T0$  de acordo com a equação 7.17, sendo  $T_{trigger}$  o valor pretendido para a periodicidade do envio dos sinais e  $T_{timer}$  o valor do período do *timer*.

$$T0 = \frac{T_{trigger}}{T_{timer}} \quad (7.17)$$

Para a configuração do desfasamentos, uma lógica semelhante é aplicada. O cálculo do valor da variável  $T2$ , por exemplo, que representa o desfasamento entre os *triggers* B e C, pode ser feito seguindo a equação 7.18, sendo  $T_{desf}$  o valor pretendido para o desfasamento e  $T_{timer}$  o valor do período do *timer*.

$$T2 = \frac{T_{desf}}{T_{timer}} + T1 + 1 \quad (7.18)$$

O *software* implementado nos sistemas de emissão presentes nas estações tem como objetivo, após receber o sinal oriundo da placa de *trigger*, utilizar o DAC para enviar o sinal codificado com a identificação da respetiva estação.

Uma vez que a frequência de amostragem da placa de receção era de 500 kHz, o sinal criado pelo emissor idealmente possuiria também ele 500 amostras por segundo. Assim, um *timer* foi configurado para a frequência de 500 kHz e, na sua interrupção, elaborado um algoritmo responsável pela elaboração do sinal pretendido no DAC do microcontrolador, executado 500 vezes por segundo. Um excerto do código elaborado para este efeito pode ser observado no apêndice B.1.

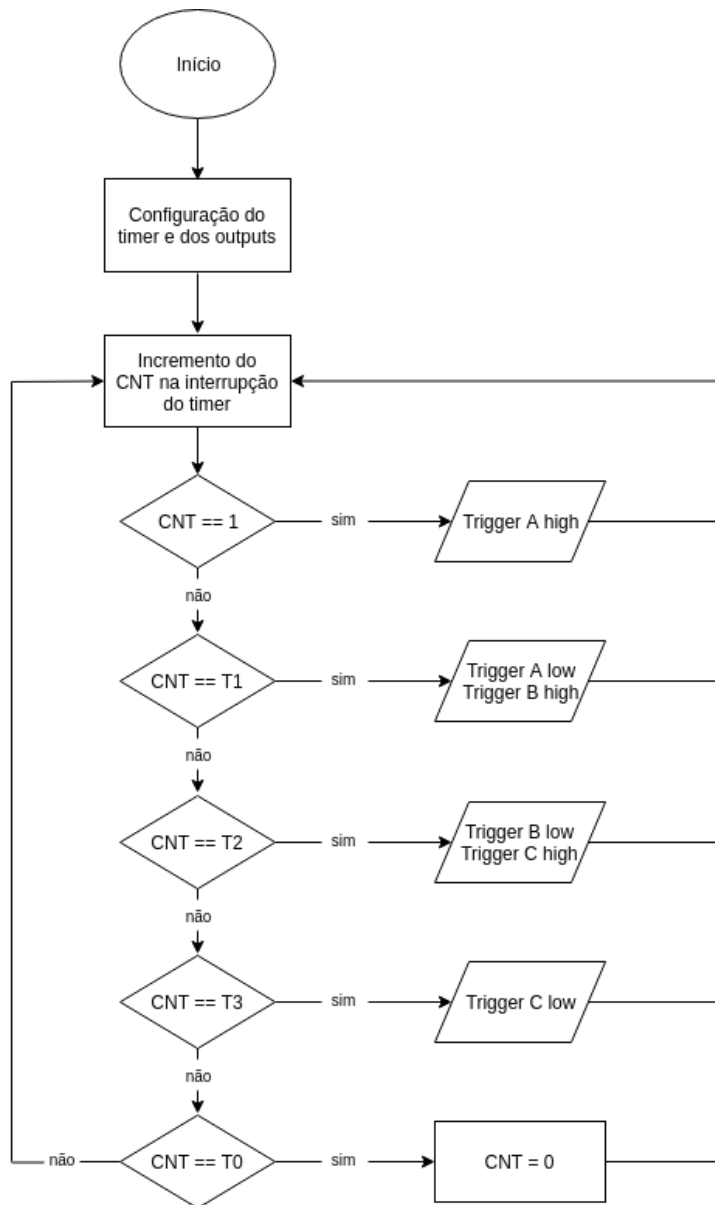


Figura 7.14: Algoritmo da placa de *trigger*

No exemplo presente no excerto, a mensagem enviada é constituída por três *pings*, espaçados pelos intervalos definidos no vetor *msg*. O *ping* aqui usado é um sinal *chirp* de 1 ms, tal como se pode concluir pelas 500 posições do vetor *ping*.

Na figura 7.15 é possível observar um diagrama que exemplifica o algoritmo implementado no sistema emissor das estações. Após a inicialização do *timer*, do DAC e da interrupção externa, sempre que é detetado o sinal de *trigger* na entrada correspondente à interrupção externa, o ciclo de envio do sinal codificado é iniciado.

Após o envio estar completo, o *timer* é desligado, ficando o sistema à espera de novo *trigger* para repetir o processo.

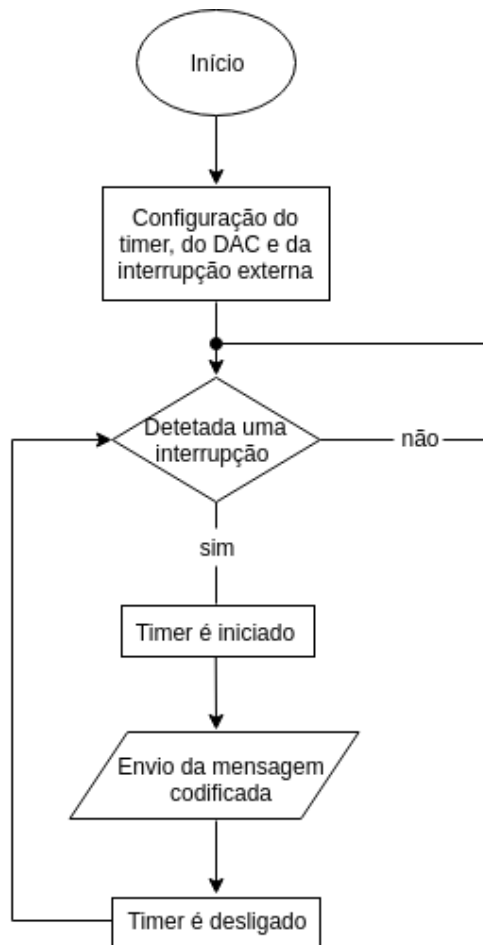


Figura 7.15: Algoritmo do sistema de emissão das estações

O *software* implementado no sistema de emissão presente no alvo tem como objetivo, após a recepção por *universal synchronous and asynchronous receiver-transmitter* (USART) da posição atual calculada (latitude, longitude e profundidade), colocar essa informação em formato binário, procedendo à sua codificação para posterior envio do sinal codificado.

Após a recepção dos dados por USART, contendo os valores de latitude, longitude e profundidade, estes dados são separados e colocados em formato binário. Dado que a mensagem binária tem sempre tamanho fixo, com o número de *bits* alocados para cada um dos valores também ele não variável, o mesmo algoritmo de codificação binária pode ser aplicado independentemente da informação recebida.

Um excerto do código elaborado para fazer a conversão dos dados para o formato binário está exposto no apêndice B.2.

Após a mensagem estar no formato binário, os valores dos diferentes *bits* são transformados em valores de intervalos, em amostras. Posteriormente, o *timer* é iniciado e a mensagem enviada, utilizando o algoritmo de envio exposto anteriormente.

Na figura 7.16 é possível observar um diagrama que exemplifica o algoritmo implementado no sistema emissor presente no alvo.

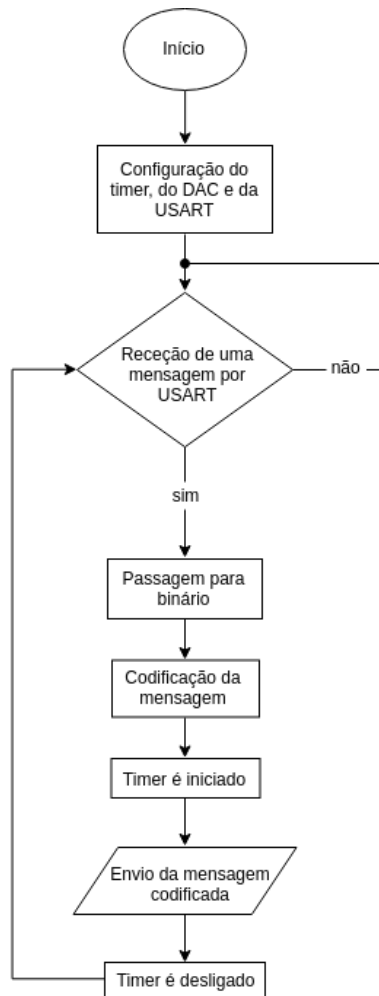


Figura 7.16: Algoritmo do sistema de emissão do alvo

## 7.4 Validação dos Sistemas

Após a implementação dos diferentes sistemas, foi realizado um conjunto de testes para validar a implementação dos mesmos. Neste conjunto de testes, pretendia-se registrar o desempenho conjunto dos módulos de condicionamento elaborados e das placas NUCLEO, com o *software* desenvolvido.

Nesta secção são expostos *prints* de osciloscópio que ilustram o correto funcionamento da placa de *trigger*, do sistema de emissão, nas suas duas variantes, e do sistema de receção.

Para testar a placa de *trigger*, foi definida a periodicidade do envio dos sinais para 10 s e o desfasamento entre o envio dos sinais de 0.1 s. Na figura 7.17, estão presentes os três sinais gerados pela placa de *trigger*, em três canais diferentes do osciloscópio. É também

possível observar nas medições do osciloscópio presentes na parte inferior da imagem que o tempo que os três sinais passam no nível lógico *high* corresponde a 0.1 s, sendo esta medida também representativa do desfasamento entre envios, como foi exposto no subcapítulo 7.3.

Na figura 7.18 é possível observar a periodicidade do envio dos três sinais, registada na parte inferior da imagem, correspondente aos 10 s esperados.

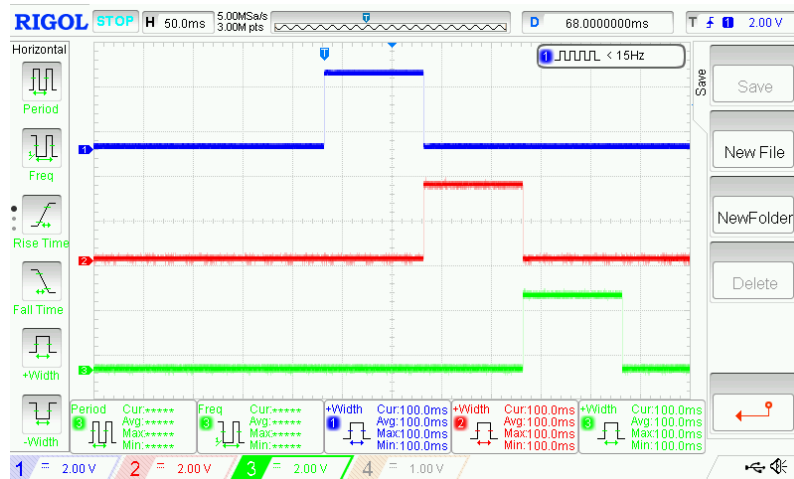


Figura 7.17: Desfasamento entre os sinais de *trigger* gerados

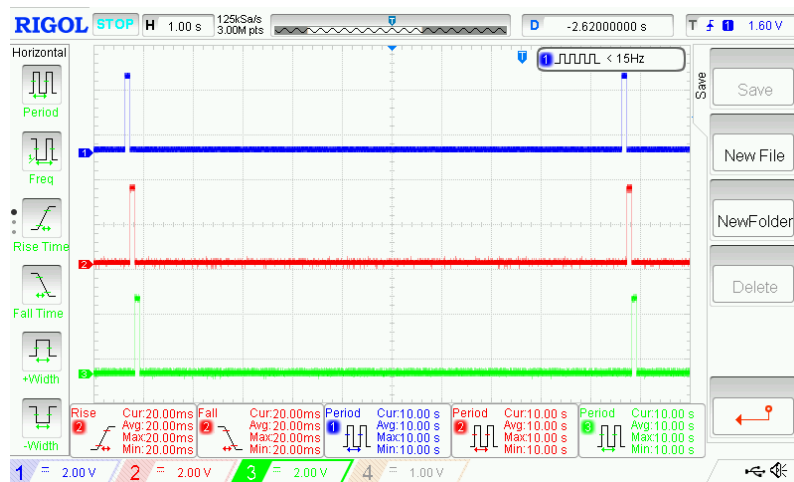


Figura 7.18: Periodicidade dos sinais de *trigger* gerados

Estes sinais de *trigger* foram depois aplicados a três sistemas de emissão diferentes, cada um representativo de uma estação.

Na figura 7.19 estão expostos os sinais gerados pelos DAC dos três sistemas de emissão, após receberem os sinais oriundos da placa de *trigger*.

Na figura 7.20 podem ser observados com maior detalhe os desfasamentos entre os sinais gerados, correspondentes a 0.1 s, o que é comprovado pela existência de 5 intervalos da escala temporal entre eles com valor de 0.02 s.

Um sinal gerado pelo sistema de emissão presente no alvo, responsável pela codificação e envio de uma mensagem recebida contendo valores de latitude, longitude e altitude, pode ser observado na figura 7.21.

Com o intuito de validar o condicionamento de sinal desenvolvido, foram realizados testes comparativos entre o sinal à entrada e à saída do condicionamento. Na figura 7.22 pode ser observada uma comparação entre um sinal *chirp* de entrada, a azul, e a saída do módulo de condicionamento de sinal, a vermelho. A saída apresenta valores de amplitude cerca de 5 vezes superiores, como se pode constatar pelos valores das duas escalas verticais, presentes no canto inferior esquerdo da imagem. Para um sinal de entrada com cerca de 3.24 V<sub>pp</sub>, o sinal de saída apresenta 16.6 V<sub>pp</sub>, como é possível observar nas medições do osciloscópio presentes na parte inferior da imagem.



Figura 7.19: Sinais gerados pelos três DAC

Por fim, nas figuras 7.23 e 7.24, está presente a relação entre um sinal de entrada (um *ping*) no sistema de recepção, oriundo do hidrofone, e o sinal de saída do módulo de condicionamento de sinal. Uma vez que o sinal oriundo do hidrofone apresenta valores de tensão muito baixos em circunstâncias normais, para esta representação um hidrofone recetor foi colocado a cerca de 5 cm do emissor. Neste cenário é possível verificar que, para um sinal de entrada no módulo de condicionamento com valores de 60 mV<sub>pp</sub> (figura 7.23, com o valor de amplitude indicado na parte inferior), o sinal na entrada do microcontrolador está limitado pelo diodo de Zener, apresentando um valor de 3.22 V<sub>pp</sub> (figura 7.24, com o valor de amplitude indicado na parte inferior).



Figura 7.20: Desfasamentos dos sinais gerados pelos três DAC

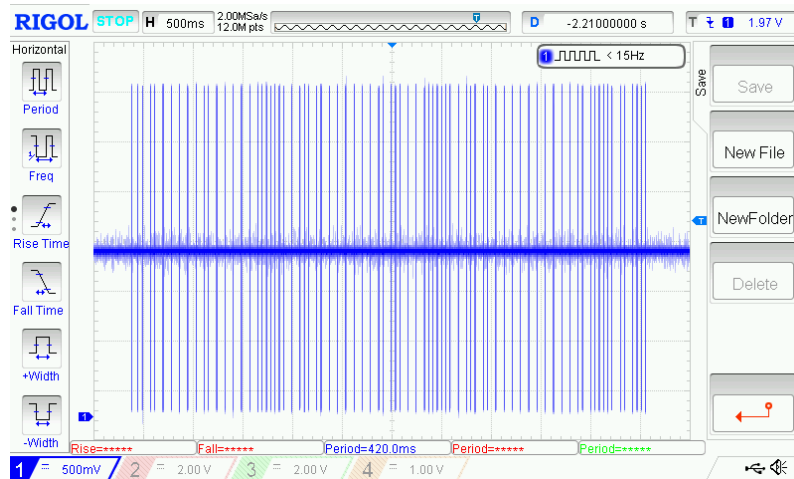


Figura 7.21: Sinal gerado com a mensagem codificada

## 7.5 Implementação em ROS

Para possibilitar a integração deste sistema com outros existentes no veículo e a aquisição de medições em tempo real, os diferentes algoritmos relativos ao processo de detecção e cálculo de posição foram implementados em ROS usando a linguagem de programação *Python*.

O ROS é um *middleware* para o desenvolvimento de aplicações robóticas. Possui um conjunto de ferramentas, bibliotecas e convenções que permite uma abstração de *hardware*, controlo de dispositivos de baixo nível, passagem de mensagens entre diferentes processos, entre outras funcionalidades. Uma aplicação ROS é constituída por um conjunto de nós representando processos em execução, que publicam e subscrevem informação num dado tópico num paradigma de *publish-subscribe* [54].

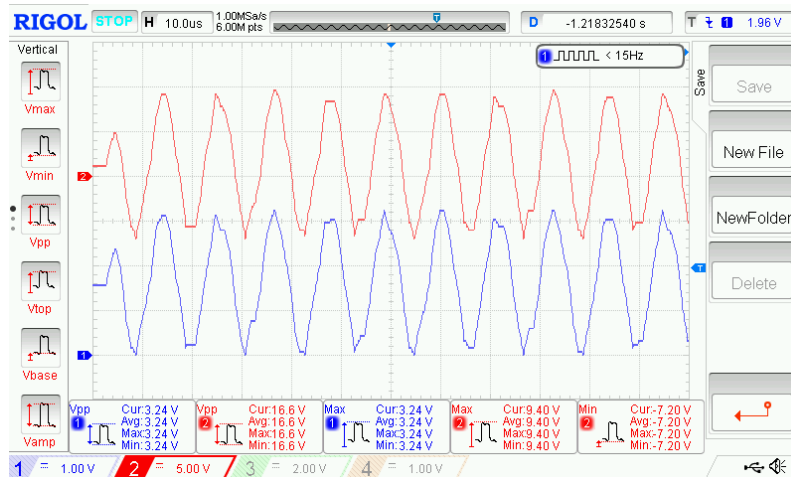


Figura 7.22: Amplificação do sinal gerado

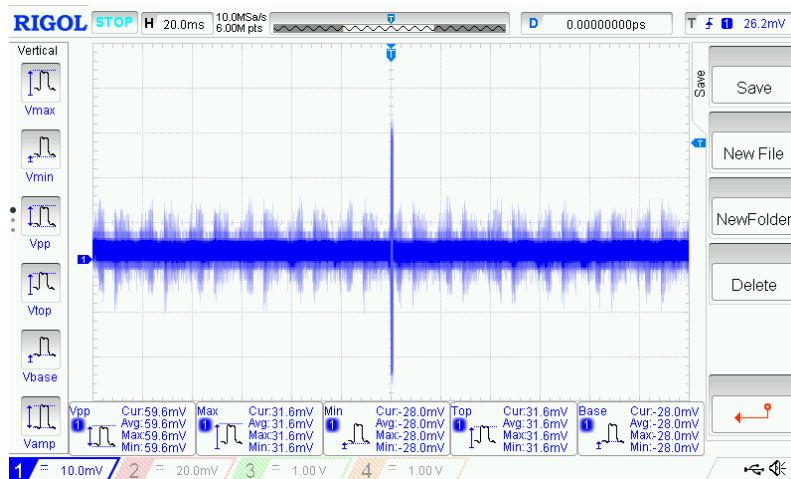


Figura 7.23: Ping na saída do hidrofone

As principais bibliotecas clientes de ROS (*C++*, *Python* e *Lisp*) são suportadas pelo sistema operativo *Ubuntu Linux*, sendo por isso esse o sistema operativo aqui usado.

O sistema de gravação desenvolvido, responsável pela aquisição dos dados oriundos de um hidrofone a uma taxa de 500000 amostras por segundo, envia os dados de acordo com o protocolo UDP, através de um cabo de *ethernet*. No microcontrolador, os dados são adquiridos usando funções de *Direct Memory Access* (DMA), preenchendo posteriormente um *buffer* de envio. Esta estratégia permite que não ocorra perda de dados mesmo nos momentos em que a mensagem está a ser enviada.

O processamento em ROS dos dados enviados pelo microcontrolador está distribuído por vários nós, estando na figura 7.25 representado um diagrama com os nós utilizados e os tópicos em que estes publicam.

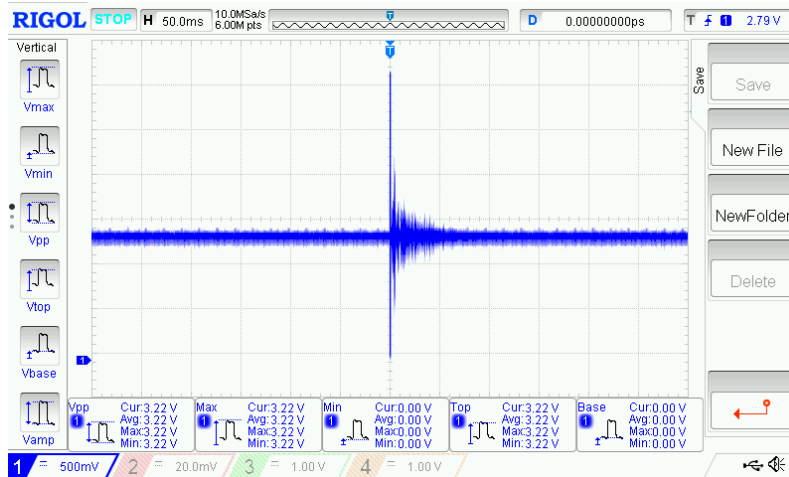


Figura 7.24: Ping na saída do módulo de condicionamento de sinal

Um primeiro nó, denominado de *ad\_receiver*, é responsável por recolher as mensagens UDP e transformar a informação numa mensagem *standard* de ROS do tipo *Joy*, para assim possibilitar uma fácil utilização do sistema de gravação por qualquer sistema que dele necessite. Após a transformação estar concluída, a informação é publicada no tópico *ad\_measurements*, que é subscrito pelo nó *message\_listener*. Este nó tem a função de acumular as amostras oriundas do ADC e assim criar uma janela deslizante para processamento de dados. Uma vez que o método de identificação dos sinais se baseia no uso de uma correlação, um sinal que se encontre numa zona de transição entre janelas poderia não ser encontrado. Com esta estratégia de janela, uma parte final dos dados existentes na janela anterior é também processada pela janela seguinte, sendo posteriormente eliminadas possíveis deteções repetidas. A informação oriunda deste nó é publicada no tópico *windowValues* e em ficheiros independentes, possibilitando a criação de *logs* da atividade do sistema.

Um novo nó, denominado de *hyper\_loc*, subscrive ao tópico *windowValues* e aplica os algoritmos de receção aos dados recebidos. Um diagrama exemplificativo do algoritmo geral de receção implementado neste nó pode ser observado na figura 7.26.

Numa primeira instância, a janela de dados recebida é correlacionada com o *ping* esperado. Posteriormente, o Algoritmo 3, presente no subcapítulo 5.2, é aplicado. Nesta fase os possíveis *pings* encontrados são guardados com uma referência temporal, para assim ser possível não só calcular os desfasamentos de chegada como também eliminar possíveis *pings* repetidos. Aqui são também eliminados *pings* que, devido à sua posição temporal, já não possam fazer parte de uma nova medição.

No passo seguinte, os *pings* capturados são analisados de forma a encontrar os sinais esperados e enviados pelos emissores. Caso se verifique a presença dos três emissores num conjunto de *pings* guardado, é registado o intervalo de chegada entre eles.

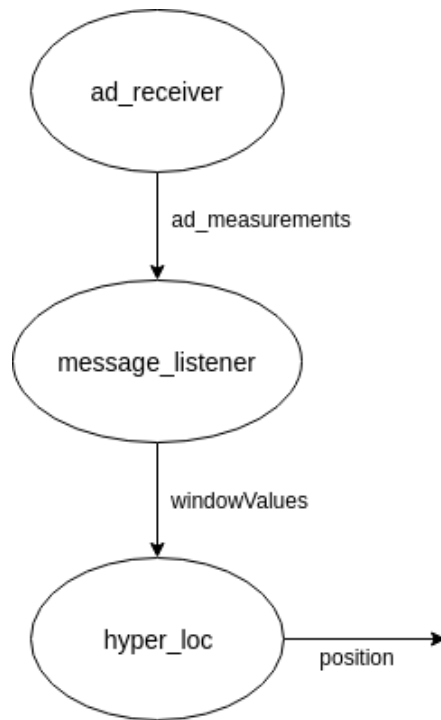


Figura 7.25: Nós de ROS utilizados

Quando os intervalos são obtidos, é aplicado o algoritmo de localização hiperbólica exposto no capítulo 6. Com a informação do sensor de pressão, das posições dos emissores e com o desfasamento medido de chegada dos sinais, o cálculo da posição do alvo torna-se possível. Após a posição ser calculada, o filtro de Kalman é atualizado com a nova observação. Aqui é também feita uma análise da ocorrência de possíveis falhas de leitura, comparando o registo temporal da medição realizada com a última que foi considerada pelo filtro. Caso se verifique que existiram leituras falhadas, são realizadas iterações do filtro apenas com fase de previsão para cada uma dessas leituras falhadas.

Por fim, após o filtro ser atualizado, a posição estimada é publicada no tópico *position*, juntamente com os valores de desfasamento medidos e a data e hora da leitura. Caso seja necessário o envio de dados para a superfície, isto é, a última leitura representar um número de medições múltiplo da periodicidade de envio configurada, a última posição é enviada por comunicação série para o sistema emissor, que a vai codificar num sinal acústico e enviar para a superfície, tal como exposto no subcapítulo 7.3. O momento de envio foi assim escolhido, imediatamente a seguir à receção de um sinal oriundo dos emissores, com o intuito de evitar ao máximo um congestionamento do meio com vários sinais acústicos distintos.

A implementação em ROS do algoritmo de descodificação da mensagem da posição, necessária para o sistema de receção presente à superfície, foi semelhante à implementação apresentada para o recetor existente no alvo. No entanto, o tópico *windowValues* é agora

subscrito por um novo nó com o nome de *msg\_read*.

O algoritmo de descodificação segue a mesma lógica do algoritmo exposto na figura 7.26. Após os *pings* estarem identificados, é aplicado o Algoritmo 4, apresentado no subcapítulo 5.3, que analisa os *pings* armazenados e descodifica a mensagem contendo os valores de latitude, longitude e profundidade do alvo. No momento em que uma mensagem é descodificada a partir de um conjunto de *pings*, esses *pings* são eliminados. Posteriormente, a mensagem descodificada é publicada no tópico *position* e guardada no *log* específico.

Após a realização dos primeiros testes à medição dos desfasamentos em ambiente laboratorial, foi observado que as medições da diferença de chegada dos três sinais apresentavam um erro maior do que o inicialmente esperado e resultante da aplicação do algoritmo de detecção dos *pings* exposto no subcapítulo 5.2. No entanto, quando observados os desfasamentos entre *pings* referentes ao mesmo emissor, estes apresentavam erros bastante inferiores e mais de acordo com os erros esperados pelo uso do algoritmo de detecção. Tendo isto como base, concluiu-se que este erro observado na medição dos desfasamentos seria maioritariamente causado por possíveis diferenças nos hidrofones usados para cada emissor, assim como por distúrbios à boa propagação do sinal que a disposição escolhida para os emissores pudesse causar. A influência da escolha da localização dos emissores é particularmente notória num cenário como refrações acentuadas, como no tanque do laboratório.

Tendo isto em conta, foi desenvolvido um algoritmo de calibração, também ele implementado em ROS, com o intuito de possibilitar a estimação do *offset* associado às medições, para assim os valores medidos serem ajustados futuramente. Este algoritmo utiliza dados reais das posições dos emissores e do alvo (que durante o processo de calibração, deverá estar estacionário) para calcular as diferenças entre os desfasamentos medidos e os esperados, guardando essas diferenças para posterior uso por parte do algoritmo de posicionamento.

Numa fase inicial do método de calibração, o algoritmo de posicionamento é parcialmente executado, sendo apenas medidos os valores de desfasamento de chegada dos sinais enviados pelos três emissores. Este processo é depois repetido por um número de vezes a definir pelo utilizador. Quando esse número de medições é atingido, é calculado um valor médio dos desfasamentos medidos. Posteriormente, e usando o valor real da posição do alvo, os desfasamentos esperados são calculados e comparados com os medidos. Os erros estimados são depois guardados, para serem usados pelo algoritmo de receção, presente na figura 7.26, compensando assim as medições obtidas.

Um diagrama com o algoritmo de calibração pode ser observado na figura 7.27.

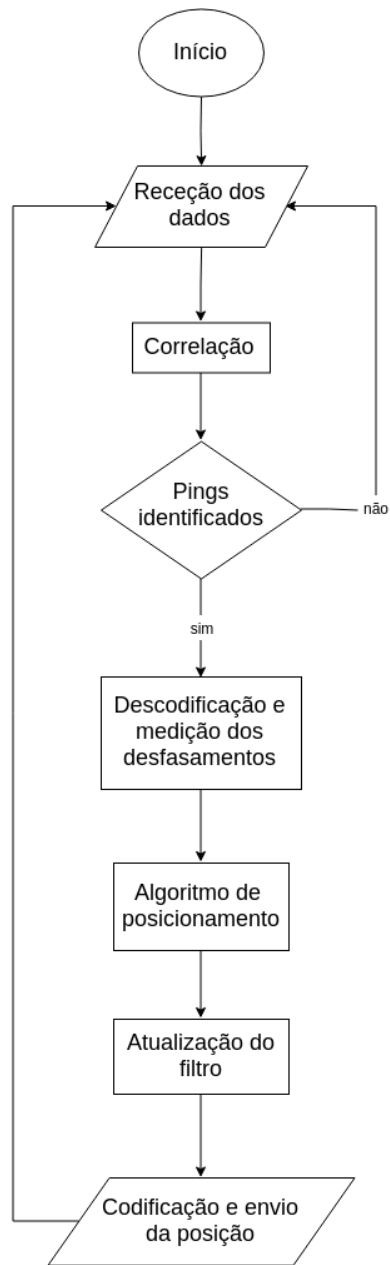


Figura 7.26: Algoritmo geral de receção

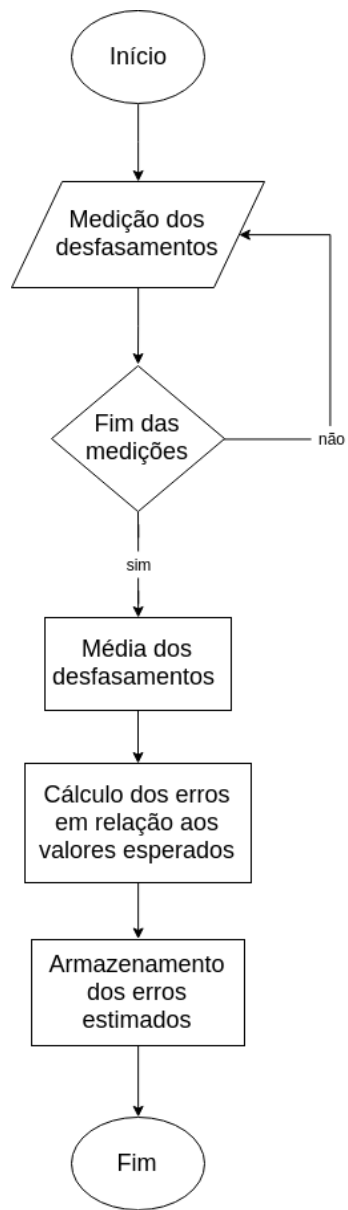


Figura 7.27: Algoritmo de calibração

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

# Capítulo 8

## Resultados

Neste capítulo são apresentados, numa fase inicial, os resultados obtidos com os testes realizados em ambiente de laboratório e em ambiente de rio. Posteriormente, é apresentada a simulação desenvolvida e expostos os testes nela realizados.

### 8.1 Testes em Laboratório

Com o intuito de avaliar a capacidade do sistema desenvolvido em estimar corretamente a posição de um alvo e de enviar uma mensagem codificada, foi realizado um conjunto de testes no tanque do laboratório (figura 8.1). De notar que ao longo de todo o período de desenvolvimento deste trabalho vários testes foram sendo realizados, sendo no presente subcapítulo apenas apresentados os testes realizados na fase final do desenvolvimento. Numa fase inicial serão apresentados os testes feitos para avaliar a exatidão do posicionamento do sistema, sendo depois exposto o resultado ao teste de distância feito à receção de uma mensagem codificada.

Com os testes de posicionamento pretendia-se também estudar as possíveis causas de erro nas medições, tendo para isso sido criados vários cenários no tanque do laboratório, com o alvo em diferentes posições e profundidades. Para estudar também qual seria a disposição para os emissores que melhor otimizaria os resultados, as posições do alvo foram repetidas para duas disposições diferentes dos emissores.

Na figura 8.2 podem ser observadas, à escala, as sete posições de alvo testadas, representadas por círculos amarelos, e as duas distribuições testadas para os emissores, representadas por triângulos negros (a primeira configuração) e por triângulos vermelhos (a segunda configuração). As sete posições foram escolhidas tendo em conta as características do tanque, procurando sempre afastar o recetor das paredes do mesmo, possibilitando assim o estudo dos erros das medições quando o alvo se encontra em diferentes posições em relação à *baseline* de emissores. Os testes realizados para uma distribuição de *baseline* em linha foram replicados para uma distribuição em triângulo. Considerando como origem do

referencial o canto inferior esquerdo do tanque, as posições do alvo e dos emissores podem ser observadas na tabela 8.1.

Recorrendo a ferramentas existentes no laboratório, a temperatura da água foi medida, permitindo assim estimar com relativa exatidão a velocidade de propagação dos sinais acústicos no meio. Antes das medições com as duas *baselines* diferentes se iniciarem, o sistema foi calibrado com o algoritmo de calibração, ficando esta calibração a ser usada no decorrer dos diferentes testes para a mesma distribuição de *baseline*.

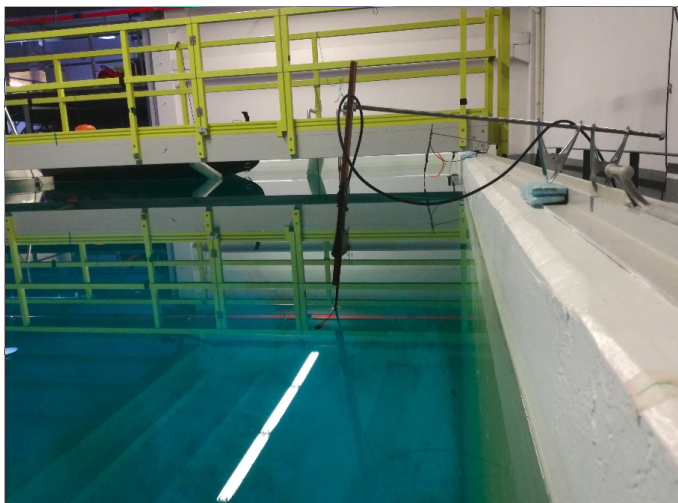


Figura 8.1: Testes no tanque do laboratório

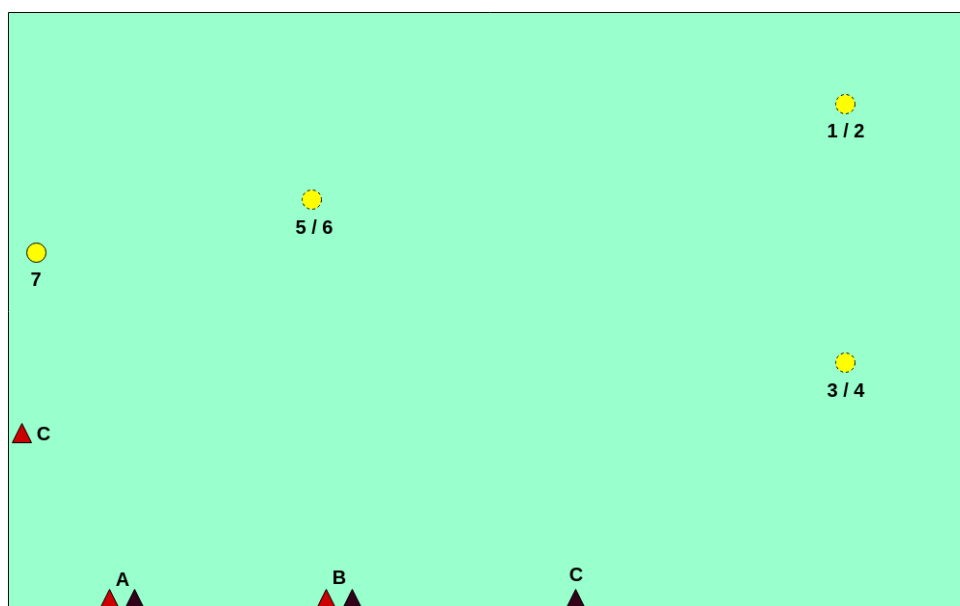


Figura 8.2: Disposição dos testes no tanque do laboratório

Por uma questão de simplificação da apresentação dos resultados, erros relativos a posições com características semelhantes são aqui agrupados nas mesmas tabelas. Nas tabelas são apresentados os valores absolutos dos erros médios (em amostras e em metros) entre os valores esperados e os medidos, tanto para os desfasamentos entre sinais oriundos de emissores diferentes e usados para calcular a posição do alvo ( $\Delta_{12}$  e  $\Delta_{13}$ ), como para os desfasamentos medidos entre *pings* relativos ao mesmo sinal, usados para identificar os diferentes sinais ( $\Delta_{id}$ ). Como neste sistema cada emissor envia um sinal com três *pings*, é possível medir seis intervalos entre *pings* relativos ao mesmo emissor. Por uma questão de simplificação, o valor apresentado nas tabelas é referente ao erro médio apresentado pelos seis desfasamentos. Uma vez que os valores de erro médio permitem apenas estudar a exatidão das medições, são também apresentados os valores do desvio padrão, em metros, para cada uma das medidas, para exemplificar a precisão dos resultados obtidos. As tabelas estão expostas no apêndice C.

Tabela 8.1: Posição (em metros) dos hidrofones no tanque

Hidrofone	Px	Py	Pz
1	8.69	2.57	-0.30
2	8.69	2.57	-2.00
3	8.69	5.35	-0.30
4	8.69	5.35	-2.00
5	3.17	4.56	-0.30
6	3.17	4.56	-2.00
7	0.29	3.71	-0.30
A	1.31	0.12	-0.30
B	3.57	0.12	-0.30
C <sub>1</sub>	5.89	0.12	-0.30
C <sub>2</sub>	0.14	1.84	-0.30

Para a primeira configuração, o valor máximo de erro na medição da diferença de distâncias entre o alvo e dois emissores foi de 0.13 m, o valor médio foi de 0.08 m e o valor mínimo de 0.03 m. O valor médio do desvio padrão em todas as medições foi de 0.06 m.

Para a segunda configuração, o valor máximo de erro na medição da diferença de distâncias entre o alvo e dois emissores foi de 0.20 m, o valor médio foi de 0.09 m e o valor mínimo de 0.03 m. O valor médio do desvio padrão em todas as medições foi de 0.05 m.

Ao observar os resultados obtidos, é possível concluir que, nos cenários elaborados, a distribuição da *baseline* e as posições escolhidas para o alvo não influenciam claramente os erros nas medições, apesar de os valores de erro apresentados pela configuração em linha serem ligeiramente mais baixos. Por outro lado, é perceptível ao observar as tabelas a tendência dos erros associados aos intervalos de identificação dos sinais em serem inferiores aos erros estimados para o cálculo do desfasamento entre sinais de emissores distintos. Os erros que aparecem associados à medida dos desfasamentos entre *pings* oriundos do mesmo

sinal (intervalos de identificação) traduzem o erro imposto pelo algoritmo de detecção dos *pings* que, apesar de obter resultados bastante satisfatórios, não consegue evitar estar sujeito às irregularidades da comunicação acústica subaquática. Os erros associados às medições dos desfasamentos entre emissores são não só influenciados pela irregularidade do meio, como também por possíveis refrações, diferenças no *hardware* emissor, distribuição da *baseline*, erros humanos de medição das distâncias reais, entre outros. De notar também que, sem o algoritmo de calibração, estes erros associados aos desfasamentos entre sinais distintos seriam superiores, conseguindo assim a calibração prévia do sistema minimizar alguns dos fatores anteriormente expostos.

Uma vez que a exatidão do algoritmo de posicionamento apresentado no capítulo 6 está muito relacionada com as dimensões da *baseline* utilizada, um cenário como o construído no tanque do laboratório não é o adequado quando se procura um bom posicionamento do alvo. Devido às dimensões muito reduzidas da *baseline*, um pequeno erro na medição dos desfasamentos origina um erro significativo no cálculo do ponto de convergência das três hipérbolas. Devido a limitações tanto de *hardware* como de espaço útil para a realização de testes, não foi possível realizar testes práticos com *baselines* de grandes dimensões. No entanto, uma vez que durante o decorrer do desenvolvimento deste trabalho foi desenvolvida uma simulação para testar os algoritmos criados (simulação essa exposta no subcapítulo 8.3), que possibilita não só a manipulação ilimitada da *baseline* como também dos erros associados às medições, o estudo da exatidão do posicionamento a partir dos erros de medição verificados pode na mesma ser realizado. De notar que cenários diferentes e menos controlados originarão erros diferentes nas medições mas, uma vez que estes são difíceis de prever, os valores usados para representar os erros nas medições irão ser os observados no tanque do laboratório, que já de si apresenta um cenário bastante extremo, devido à grande quantidade de reflexões nele existente.

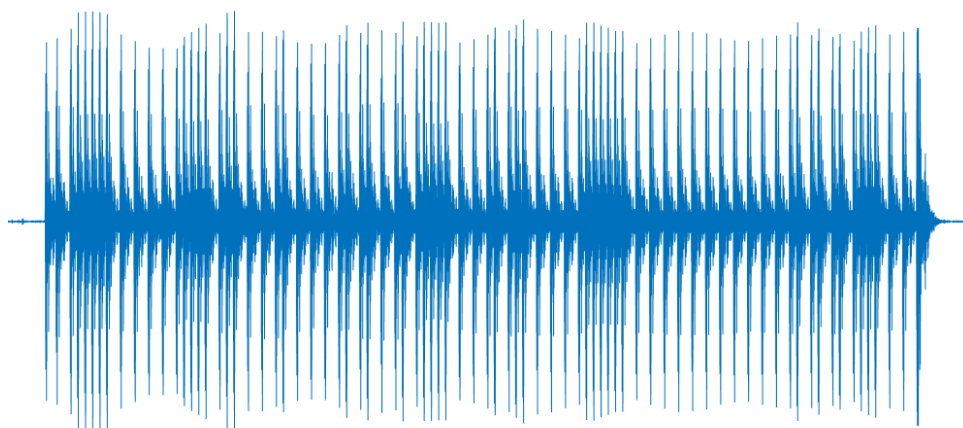


Figura 8.3: Mensagem codificada adquirida pelo recetor

Em paralelo com os testes de posicionamento, foram realizados no tanque do laboratório testes ao envio e à receção da mensagem codificada, com uma taxa de sucesso de interpretação a rondar os 100 %. Desta forma, foi possível validar que a mensagem era enviada e decodificada a uma distância de 8 m. Na figura 8.3 é possível ser observada a correlação da mensagem com 80 *pings* captada pelo recetor a 8 m do emissor.

## 8.2 Testes de Campo

Aproveitando uma missão de campo do INESC TEC a Avintes, em Vila Nova de Gaia, foram realizados testes para avaliar a capacidade do sistema em transmitir um sinal acústico a grandes distâncias. Para tal, um emissor foi colocado no rio Douro, junto à margem, enquanto que um recetor foi colocado a bordo de um barco usado na missão. O emissor foi programado para enviar periodicamente uma mensagem codificada, enquanto o barco realizava um percurso no rio. Mais tarde, após aceder aos dados captados pelo recetor e aos *logs* de GPS do barco, foi possível concluir que a mensagem consegue ser captada a pelo menos 1.2 km de distância, uma vez que foi este o ponto mais distante da trajetória feita pelo barco ao longo do dia. Devido à qualidade do sinal recebido a essa distância, pode-se concluir que uma distância maior será possível de ser validada. No entanto, devido à geografia do rio e à necessidade de afastar em demasia o barco da base de operações, não foram realizados testes para maiores distâncias.

Na figura 8.4 é possível observar uma imagem de satélite ilustrativa do teste a uma distância de 1.2 km, onde as posições do recetor (junto à margem) e do emissor (no meio do rio) se encontram assinaladas.



Figura 8.4: Teste de distância (imagem do retirada do Google Maps)

### 8.3 Simulação

Para verificar a viabilidade do algoritmo, foi desenvolvida uma simulação recorrendo ao *software Matlab*.

Esta simulação permite a criação de diferentes cenários, com posições configuráveis para todos os elementos, gerando posteriormente uma trajetória aleatória para um alvo na área de simulação criada. De seguida, os envios periódicos de sinal por parte das estações emissoras são simulados, criando um registo dos desfasamentos de chegada dos sinais oriundos das diferentes estações captados pelo alvo ao longo do tempo. Com estes dados, o algoritmo pode ser aplicado e os resultados comparados com a trajetória original criada pela simulação.

Os parâmetros configuráveis na simulação criada estão presentes na tabela 8.2.

Tabela 8.2: Parâmetros configuráveis na simulação

Parâmetros	Unidades
dimensão $(x, y, z)$	m
posições iniciais $(x, y, z)$	m
velocidade inicial $(v_x, v_y, v_z)$	m/s
tempo de simulação	s
step	s
variação máxima de velocidade	m/s
periodicidade de envio	s
erro do sensor de pressão	m
erro da medição dos desfasamentos	s
probabilidade de falhas de receção	--

Numa fase inicial, a dimensão da zona de simulação é definida, sendo criado um espaço de simulação com dimensão escolhida pelo utilizador. As posições iniciais das estações e do alvo podem também ser modificadas, assim como a velocidade inicial do alvo que, ao contrário das estações, irá-se mover ao longo da simulação. Outros parâmetros configuráveis que estão associados ao início da simulação são o tempo de simulação e o *step*, que representa o número de vezes por segundo em que a posição do alvo irá ser simulada.

Posto isto, e com base na posição e velocidade inicial do alvo, a sua trajetória é gerada. A cada iteração (ou *step*) da simulação, é gerado um valor de variação de velocidade aleatório para cada dimensão, restringido pela variação máxima definida pelo utilizador. Usando a equação 8.1 é possível gerar a nova posição no final da iteração, em que  $X, Y$  e  $Z$  representam os novos valores de posição do alvo,  $V_X, V_Y$  e  $V_Z$  as novas velocidades,  $\Delta t$  o *step*,  $x, y$  e  $z$  a posição anterior do alvo,  $v_x, v_y$  e  $v_z$  as velocidades prévias, e  $\Delta v_x, \Delta v_y$  e  $\Delta v_z$  os valores de variação de velocidade para as três dimensões.

No caso do alvo se estar a aproximar dos limites da área de simulação, os seus valores de velocidade são automaticamente manipulados para que este comece a descrever uma trajetória que o afaste do limite em questão.

$$\begin{bmatrix} X \\ Y \\ Z \\ V_X \\ V_Y \\ V_Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ v_x + \Delta v_x \\ v_y + \Delta v_y \\ v_z + \Delta v_z \end{bmatrix} \quad (8.1)$$

Na figura 8.5 está presente um exemplo de uma trajetória gerado em simulação, onde é possível também observar o posicionamento das três estações na superfície. Os valores nos eixos das figuras com representação de trajetórias simuladas aqui presentes (da figura 8.5 à figura 8.8) encontram-se em metros.

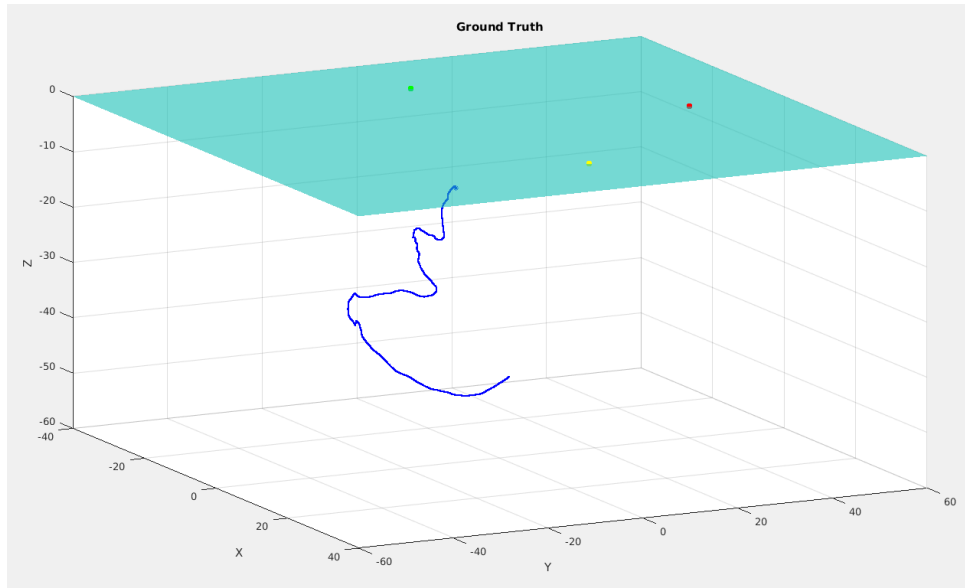


Figura 8.5: Exemplo de trajetória gerada em simulação

Após a trajetória estar criada, as medições dos desfasamentos são simuladas. De acordo com o valor definido para a frequência de envio de sinal por parte dos emissores, para certas posições do alvo, as distâncias em relação às três estações são calculadas, usando a equação 6.17. Após as três distâncias estarem calculadas, os valores simulados para desfasamento de chegada entre os sinais podem ser também eles ser calculados, usando a equação 8.2. Como exemplo, nela está representado o cálculo do desfasamento entre a chegada dos sinais oriundos das estações *A* e *B* a uma posição arbitrária do alvo, em que  $t_a$  e  $t_b$  representam os instantes de chegada dos sinais oriundos da estação *A* e da estação *B*, respetivamente,

$v$  a velocidade de propagação do sinal no meio e  $dist_a$  e  $dist_b$  as distâncias em relação às respectivas estações.

$$t_a - t_b = \frac{dist_a - dist_b}{v} \quad (8.2)$$

Posteriormente, e usando os dados simulados de defasamentos, o algoritmo de posicionamento é aplicado. Tanto às medidas de defasamentos como às medidas de profundidade (oriundas de um sensor de pressão simulado) é adicionado ruído normalmente distribuído com média nula e com desvio padrão configurável nos parâmetros iniciais, permitindo assim uma avaliação do quanto o desempenho do algoritmo é afetado com os diferentes erros de medição.

Na figura 8.6 é possível observar o resultado da aplicação do algoritmo aos defasamentos simulados para uma determinada trajetória. Aplicando apenas o algoritmo de B. T. Fang, as medições dão origem às duas trajetórias representadas. No entanto, com as alterações efetuadas ao algoritmo original, expostas no subcapítulo 6.1, é possível separar a trajetória pretendida (representada a verde) da trajetória simétrica e incorreta (representada a vermelho).

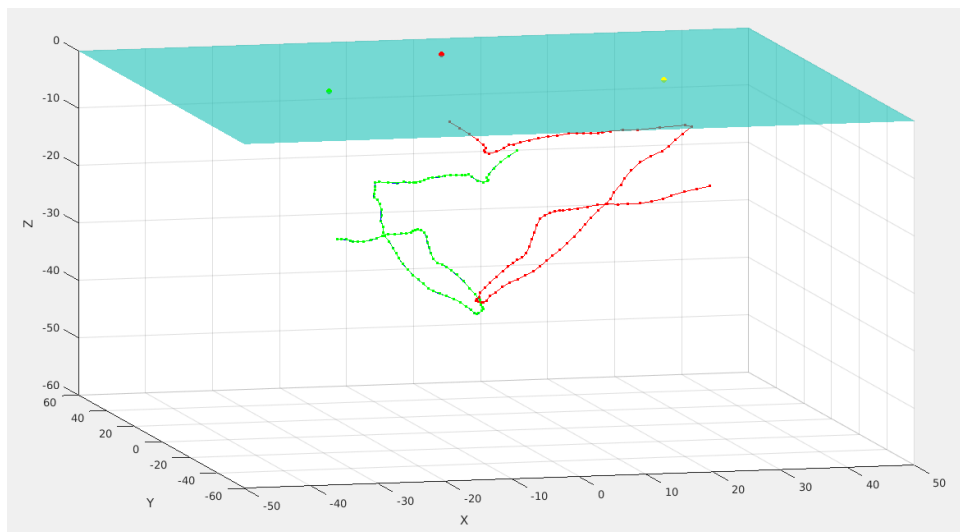


Figura 8.6: Exemplo da aplicação do algoritmo para recriação da trajetória gerada

Na figura 8.7 está representada a diferença entre uma trajetória sem ruído associado gerada pelo algoritmo (a azul) e a trajetória recriada pelo algoritmo (a vermelho), após ser acrescentado ruído nas medições de defasamento e nas de profundidade. Para este exemplo em particular, às medições de profundidade e de defasamento foi adicionado ruído normalmente distribuído com um desvio padrão de 0.1 m e de 200  $\mu$ s, respetivamente.

Dando seguimento ao estudo apresentado no subcapítulo 8.1, o ambiente de simulação criado foi usado para avaliar o impacto dos erros de medição obtidos nos testes de laboratório no algoritmo de posicionamento e, conseqüentemente, na estimação da posição do alvo em cenários com *baselines* de maiores dimensões.

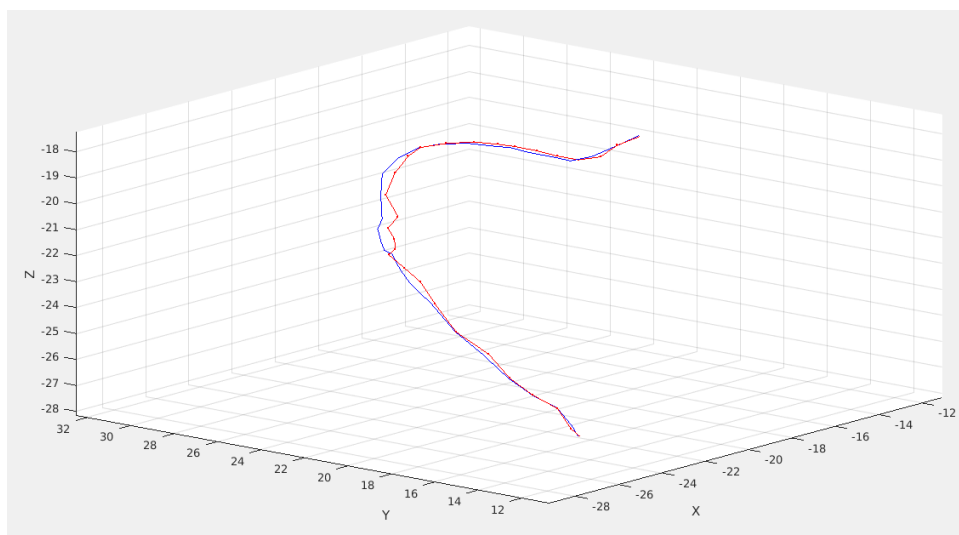


Figura 8.7: Exemplo do desempenho do algoritmo quando erros são associados às medições

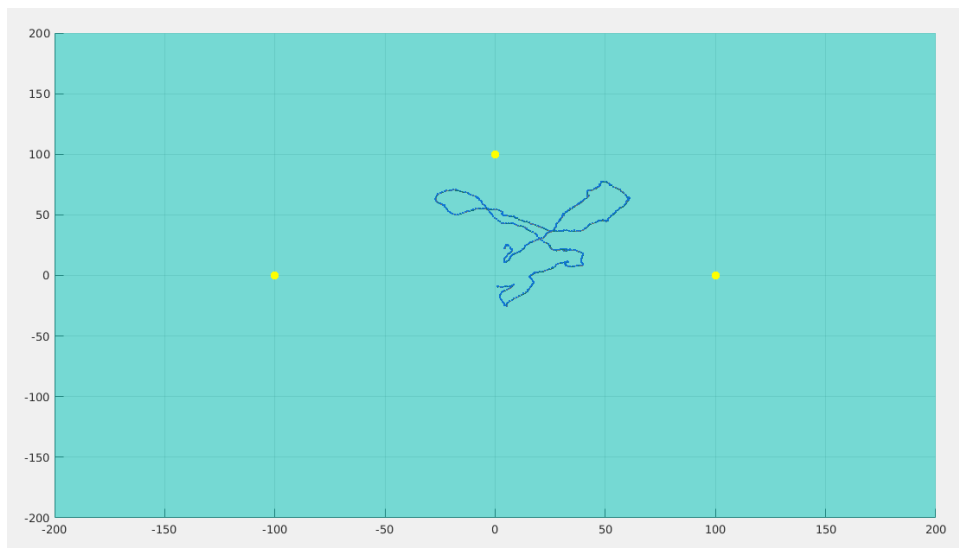


Figura 8.8: *Baseline* usada para o teste aos erros de posicionamento

Para este teste, a distribuição dos emissores foi dimensionada de forma a que estes formassem um triângulo, estando cada emissor a uma distância de 100 m da origem do referencial. Esta foi uma distribuição escolhida arbitrariamente, com o intuito de simular um cenário em que os emissores estivessem colocados em boias e sincronizados com o

auxílio de um sensor de GPS.

Na figura 8.8 é possível observar, de uma perspectiva superior, a *baseline* usada para este teste e uma trajetória simulada.

De acordo com os dados obtidos nos testes laboratoriais, o erro das medições (que na simulação criada se trata de um erro normalmente distribuído, de valor médio nulo e com desvio padrão configurável) dos desfasamentos foi configurado com sendo cerca de 0.1 m, ligeiramente superior ao erro médio de todas as medições efetuadas. O erro do sensor de pressão foi também ele configurado para 0.1 m, também ele um erro inflacionado face aos sensores de pressão normalmente usados no laboratório.

Na figura 8.9 pode ser observada a diferença ao longo da simulação, em módulo e em metros, entre a posição real (*ground truth*) do alvo e a posição estimada, usando os valores de erro de medição acima mencionados. Os resultados aqui apresentados são referentes a uma simulação em que foram efetuadas 500 medições de posição ao longo de 5000 segundos. Para esta simulação, o valor médio do erro registado ao longo das 500 leituras foi de 0.12 m.

Para obter um valor mais significativo de erro de posicionamento, foram simuladas 100 trajetórias diferentes de 5000 segundos para os mesmos valores de erro nas medições, resultando num valor médio de erro de cerca de 0.14 m.

Uma vez que é aqui usado um algoritmo de posicionamento hiperbólico, o erro de estimação da posição é também influenciado pela posição do alvo em relação aos emissores. Pretendia-se portanto, ao simular 100 trajetórias distintas, cobrir um vasto número de cenários para assim avaliar o melhor possível o comportamento do algoritmo.

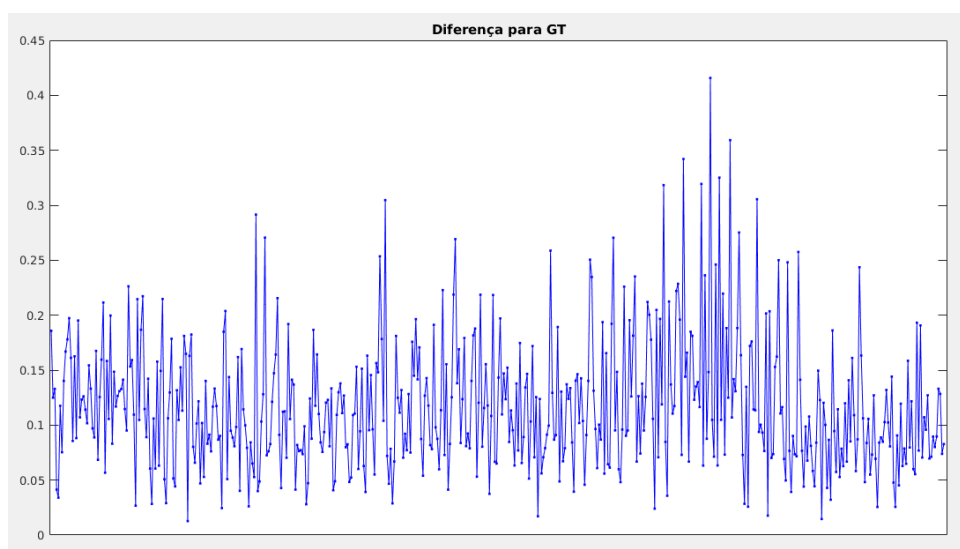


Figura 8.9: Diferença entre posição real e posição estimada

Aproveitando o ambiente de simulação criado, foi implementado um filtro de Kalman, tal como exposto no capítulo 6. O filtro tem como função não só minimizar os efeitos dos erros nas medições, possibilitando um melhoramento na estimação da posição, como também minimizar o impacto causado por falhas de leitura, em que o alvo fica sem informação relativa à sua posição atual.

Para a calibração da matriz  $R$  (exposta na equação 6.37), que representa o ruído associado às observações do sistema, um algoritmo foi desenvolvido no ambiente de simulação. Para uma determinada configuração dos parâmetros de erro, este algoritmo aplica esses erros às leituras dos sensores e executa 10 mil iterações em que o algoritmo de posicionamento é aplicado a uma posição específica. Posteriormente, é feita uma análise à variação dos resultados para as posições em  $x$ ,  $y$  e  $z$  calculadas pelo algoritmo a partir das medições sensoriais com erros. Esta análise permite extrair os valores de  $\sigma_{m_x}$ ,  $\sigma_{m_y}$  e  $\sigma_{m_z}$  associados às medições, calibrando assim a matriz  $R$ .

Como exemplo, na figura 8.10 estão expostos dois histogramas (o da esquerda representativo da posição em  $x$  e o da direita da posição em  $y$ ) resultantes da análise aos erros da estimação da posição quando é imposto ao algoritmo um erro de leitura de cerca de 0.9 m. No eixo vertical encontra-se o número de ocorrências de uma determinada medição e no eixo horizontal o valor dessa medição. Como é possível observar, a estimação do valor da posição do alvo em  $y$  é a mais sensível aos erros de medida, apresentando um desvio padrão superior.

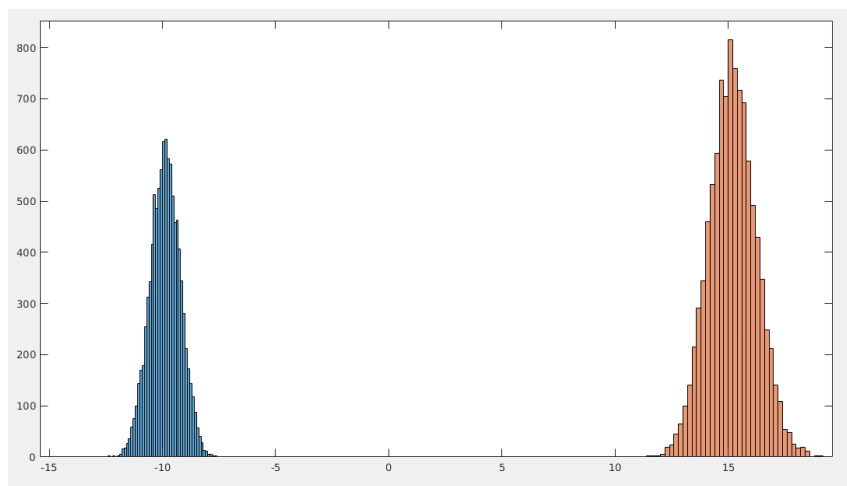


Figura 8.10: Histograma dos erros de posicionamento

Para a calibração da matriz  $Q$ , que representa o ruído associado ao sistema, os valores de desvio padrão do erro de cada estado foram manipulados, para que assim o filtro satisfizesse os três testes apresentados no capítulo 6.

Na figura 8.11, está exposto o teste feito à consistência do filtro. Aqui podem ser observados os valores das inovações referentes aos estados  $x$ ,  $y$  e  $z$  (no eixo vertical) ao

longo do tempo de simulação, assim como os limites definidos por  $\pm 2\sqrt{S_k}$ . A inovação referente ao estado  $x$  apresenta-se de acordo com os 95 % esperados. Apesar das outras duas inovações estarem com os valores todos compreendidos entre os limites, a análise deste teste pode-se considerar aceitável, uma vez que uma calibração conservadora em que os ruídos das observações são considerados superiores à realidade do sistema é uma estratégia comumente seguida. Desta forma, ao confiar menos nas medidas, irá ser obtida uma estimação que, apesar de um pouco mais ruidosa, não correrá o risco de divergir substancialmente.

Na figura 8.12, o teste da média deslizante (a vermelho) da inovação normalizada (a azul) pode ser observado ao longo do tempo de simulação, assim como os valores de  $r_1$  e  $r_2$  (a verde). Como o valor final da média deslizante se encontra entre os valores de  $r_1$  e  $r_2$ , é possível afirmar que o filtro não apresenta nenhuma tendência notória.

Por fim, na figura 8.13, é apresentado o teste ao ruído branco da inovação ao longo do tempo de simulação. Nela é possível observar a autocorrelação à inovação dos estados  $x$ ,  $y$  e  $z$ . Uma vez que os valores da autocorrelação estão distribuídos aleatoriamente à volta de 0, apresentando uma média praticamente nula, os resultados deste teste são satisfatórios.

Após ser calibrado, o filtro de Kalman foi testado em várias trajetórias simuladas, para assim a sua eficácia ser comprovada em vários cenários, sendo aqui expostos dois desses testes.

Para compreender o efeito da aplicação do filtro a uma trajetória estimada pelo algoritmo de posicionamento, é aqui apresentada na figura 8.14, a vermelho, a diferença (em módulo e em metros) entre o valor real e o valor calculado pelo algoritmo de posicionamento quando o erro de medição é de cerca de 0.2 m e, a azul, a mesma diferença após a aplicação do filtro, ao longo do tempo de simulação. Como é possível observar, a diferença para o valor real diminui ao longo da trajetória, melhorando assim a estimação da posição do algoritmo de posicionamento. Fazendo uma análise para várias trajetórias diferentes, semelhante à realizada para analisar a exatidão do algoritmo de posicionamento e com a mesma configuração de erro nas medições, foi possível verificar uma redução do erro de estimação, usando o filtro, entre 0.02 m a 0.03 m, ficando assim o posicionamento com cerca de 0.12 m de erro médio. No entanto, como esta correção é causada por algo complementar ao sistema desenvolvido e que está dependente de uma correta calibração, estes valores não são considerados na apreciação global da exatidão do sistema.

Um dos parâmetros configuráveis na simulação define a probabilidade da ocorrência de falhas de receção. Neste cenário, o alvo não consegue calcular a sua posição atual, ficando apenas com a informação da última leitura de desfasamentos. Sem a implementação do filtro, o erro de posição nesta circunstância é bastante elevado, uma vez que a posição conhecida pelo sistema é apenas a última que foi calculada. Nesta situação, o filtro de Kalman minimiza o erro de posicionamento enquanto não há novas medidas, realizando apenas a fase de *predict* e ignorando a fase de *update*. Quando se der nova leitura e

consequente *update*, o filtro deslocará a posição estimada na direção da nova leitura.

Na figura 8.15 está presente um teste igual ao apresentado na figura 8.14, mas agora com uma probabilidade de falhas de leitura de 15 %. Além das diferenças para o valor real de posição com e sem filtro, estão também indicados no gráfico os pontos onde se deu uma falha de leitura. É notória a importância da contribuição do filtro para a minimização do erro nestas alturas, uma vez que sem ele o sistema continua a achar que está na mesma posição, mesmo estando o alvo em movimento, o que leva a uma disparidade grande entre os valores calculados para a posição atual e a posição real do alvo.

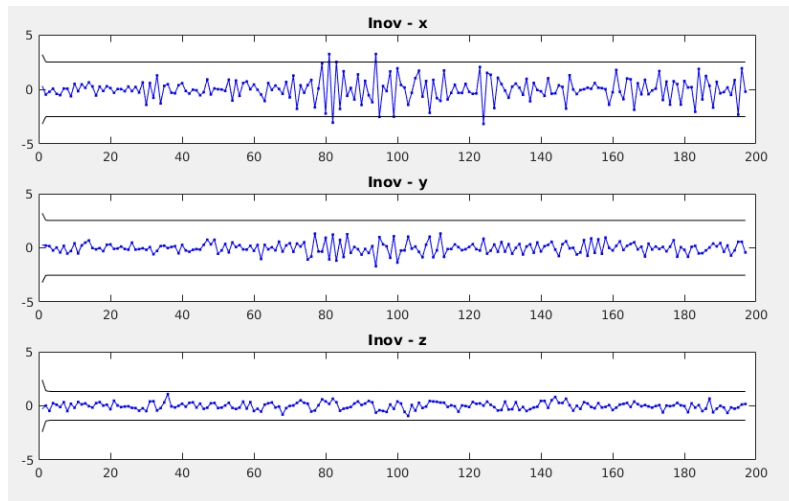


Figura 8.11: Teste à consistência dos valores da inovação

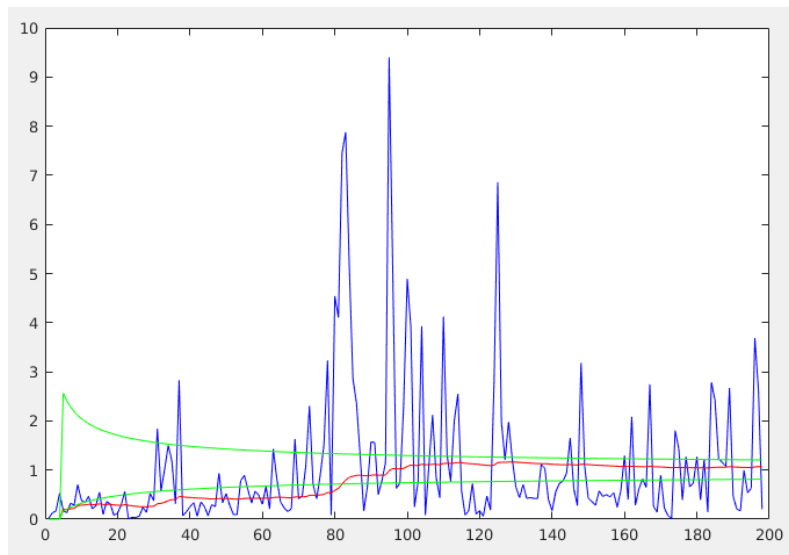


Figura 8.12: Teste à tendência do filtro

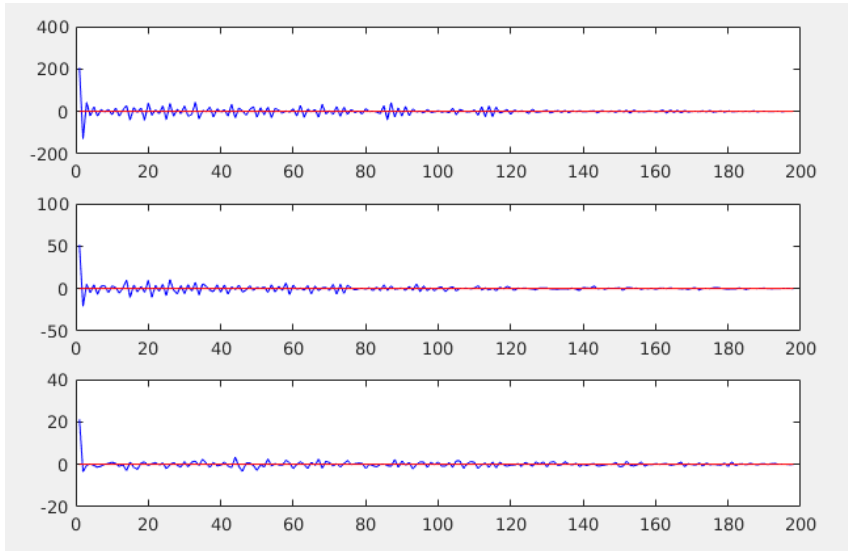


Figura 8.13: Teste ao ruído branco da inovação

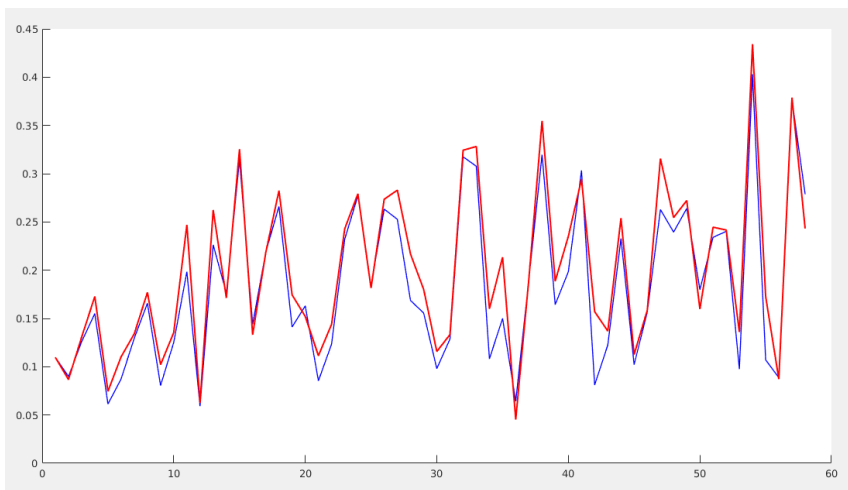


Figura 8.14: Diferença para a posição real com e sem filtro

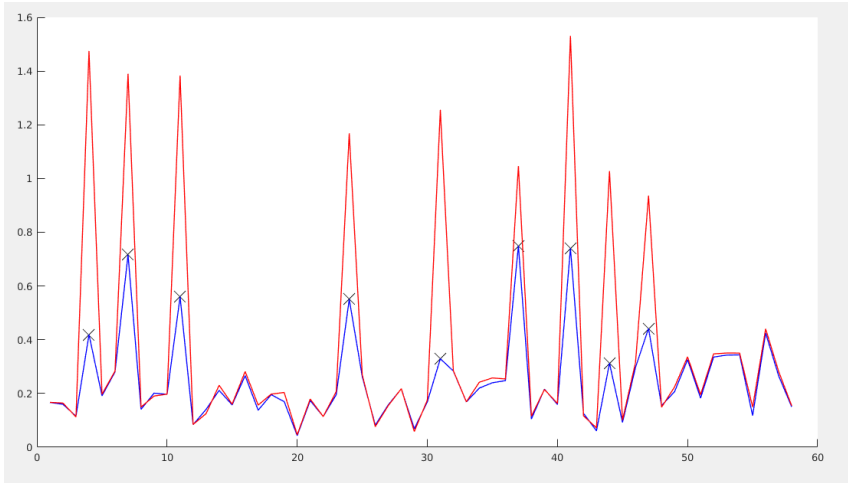


Figura 8.15: Diferença para a posição real com e sem filtro (com falhas de leitura)

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

## Capítulo 9

# Conclusão e Trabalho Futuro

### 9.1 Conclusão

Com o recente investimento em novas tecnologias numa área como a exploração subaquática, caracterizada pelo estudo de um meio com condições adversas, o foco de desenvolvimento tem-se naturalmente virado para veículos autónomos ou remotamente controlados. Este investimento despoleta o estudo de novos métodos de posicionamento para alvos subaquáticos, dada a importância que uma localização eficaz tem para um AUV ou um ROV que pretenda realizar uma missão que, dependendo das suas particularidades, pode exigir uma maior ou menor exatidão de posicionamento.

O trabalho exposto neste documento tem como objetivo o desenvolvimento de um sistema de posicionamento acústico para ser usado nos veículos subaquáticos existentes no LSA. O sistema utiliza um método de posicionamento hiperbólico, baseado na medição de TDOA, e permite o envio para a superfície de uma mensagem acústica codificada com a posição estimada pelo alvo.

No decurso do trabalho foram desenvolvidos vários subsistemas com *hardware* e *software* próprio, para efetuar a codificação, interpretação, emissão e receção de sinais acústicos.

Para a criação e codificação de um sinal acústico detentor da identificação do respetivo emissor, foi elaborado um algoritmo, abordado nos subcapítulos 7.3 e 5.3. Dada a necessidade de sincronização temporal da emissão acústica, esta é controlada por um sinal de *trigger* de sincronização, tendo o algoritmo do mesmo, apresentado no subcapítulo 7.3, sido implementado num microcontrolador de alto desempenho. Foi também desenvolvido um módulo de condicionamento de sinal, apresentado no subcapítulo 7.1, por fim a adequar o nível do sinal às características do hidrofone, maximizando assim a propagação do sinal no meio. Os resultados da interação entre a placa de *trigger* e os emissores com módulos de condicionamento de sinal podem ser observados no subcapítulo 7.4 e no capítulo 8. Foi assim elaborada com sucesso uma ferramenta que permite o envio de sinais acústicos com desfasamento configurável.

Com o intuito de possibilitar a interpretação dos sinais oriundos dos emissores e medir o seu TDOA para posterior determinação de posicionamento, foi desenvolvido um algoritmo de detecção de *pings*, apresentado no subcapítulo 5.2.

Através de uma adaptação de um algoritmo de posicionamento hiperbólico comumente usado, tornando-o assim compatível com os requisitos do projeto, o cálculo da posição de um alvo a partir dos TDOA tornou-se exequível. Foi também implementado um filtro de Kalman para minimizar os erros de estimação da posição de um alvo em movimento. O algoritmo de posicionamento e a implementação do filtro de Kalman estão documentados no capítulo 6.

Para permitir a estimação em tempo real da posição e o interface deste sistema com outros existentes num veículo subaquático, foram implementados algoritmos com recurso à *framework* de *software* ROS, tendo em vista o seu interface com os robôs do laboratório. Esta implementação está exposta no subcapítulo 7.5.

Os resultados das leituras em tempo real de vários TDOA para diferentes posições de um alvo, obtidos através da interação do *software* e do *hardware* desenvolvidos, podem ser observados no capítulo 8, onde uma análise estatística ao desempenho do sistema é apresentada. Esta análise feita no tanque do laboratório, que pode ser considerado como um ambiente de condições extremas para este teste (devido à elevada quantidade de reflexões que os sinais acústicos sofrem no seu interior), originou a obtenção de um erro médio na medição dos desfasamentos (correspondente à diferença de distâncias entre os emissores e o recetor) de cerca de 0.09 m. Pesando as características do meio e o *hardware* utilizado, este resultado é bastante satisfatório. Devido a limitações de *hardware* e de espaço no local, a *baseline* usada nestes testes era de dimensões reduzidas, o que origina um impacto grande dos erros de medição na altura de aplicação do algoritmo de posicionamento e posterior localização do alvo.

Foi também desenvolvido um ambiente de simulação, onde foi possível avaliar o desempenho do algoritmo de posicionamento e do filtro desenvolvido. Usando os valores dos erros de medição obtidos nos testes laboratoriais e manipulando a *baseline* dos emissores, um conjunto de trajetórias diferentes foram criadas e o erro de posicionamento medido, apresentando um valor médio de 0.14 m, que foi posteriormente reduzido com a implementação do filtro de Kalman. O desenvolvimento deste ambiente de simulação e os testes nele feitos estão expostos no subcapítulo 8.3.

Aproveitando o *hardware* desenvolvido para os emissores, um algoritmo de codificação acústica de uma mensagem contendo informação referente à latitude, longitude, profundidade e identificação do alvo foi elaborado, permitindo a emissão de um sinal codificado sempre que desejado. Do lado do recetor, foi implementado um algoritmo de descodificação em ROS, que permite a interpretação em tempo real do sinal recebido, reconhecendo assim a posição enviada pelo alvo. O *software* referente à emissão e receção da mensagem codificada está documentado nos subcapítulos 5.3, 7.3 e 7.5. O *software* desenvolvido foi testado

e validado em conjunto com o *hardware* dos sistemas de emissão e recepção, estando estes testes expostos nos subcapítulos 7.4 e 8.1. Os testes permitiram validar a comunicação acústica nas condições do tanque com a interpretação de múltiplas mensagens acústicas distintas a cerca de 8 m, com uma taxa de sucesso de interpretação da mensagem a rondar os 100 %.

Por fim, foi feito um teste de campo, apresentado no subcapítulo 8.2, onde o alcance de um sinal acústico gerado por um emissor desenvolvido e captado por um recetor foi avaliado. Neste teste, ficou validada a recepção da mensagem a cerca de 1.2 km de distância.

Através da interação entre os diversos sistemas desenvolvidos, o objetivo deste trabalho foi alcançado. Um sistema acústico de posicionamento foi desenvolvido de raiz, não só através da criação de *hardware* para permitir o envio e a recepção dos sinais, como através da elaboração de *software* para possibilitar a interpretação e tratamento dos mesmos. De salientar que o material usado é uma limitação aos resultados, havendo a hipótese de estes serem melhorados em troca de um maior custo de produção se, por exemplo, forem usados hidrofones de melhor desempenho. Apesar da exatidão que o sistema final iria apresentar ter sido uma incógnita até à fase final de desenvolvimento e, conseqüentemente, ter existido incerteza à volta da capacidade do sistema em cumprir o seu propósito, um erro de posicionamento como o obtido permite o uso deste sistema em inúmeros tipos de missões subaquáticas, como missões de patrulhamento, busca subaquática, estudo de *habitats*, entre outros.

Este trabalho serve assim também de base para possíveis estudos futuros que envolvam métodos de envio e recepção de sinais acústicos semelhantes, principalmente devido ao que foi apresentado ao nível de algoritmos de deteção, codificação e interpretação para estes sinais.

## 9.2 Trabalho Futuro

Apesar dos objetivos iniciais terem sido cumpridos, pode-se identificar alguns desenvolvimentos futuros para o sistema, tais como:

- Utilizar sensores GPS para obter o sinal de *trigger* para cada um dos recetores, evitando assim a necessidade de estes terem ligações entre si.
- Incorporar a análise de desvio de frequência por efeito de *Doppler*, possibilitando a estimação direta da velocidade do alvo.
- Estabelecer comunicação entre a base de operações e os emissores, possibilitando que a base tome conhecimento das suas posições em tempo real, permitindo assim o movimento livre dos emissores sem que isso afete o posicionamento correto do alvo (esta abordagem apenas cobre um cenário em que se utilize um ROV, uma vez que o

veículo perde a capacidade de reconhecer a sua própria posição, que só é conhecida pela base de operações).

- Avaliar o desempenho do sistema com testes exteriores e em ambientes operacionais com múltiplos cenários de operação e de propagação acústica.

# Apêndice A

## Módulos Desenvolvidos

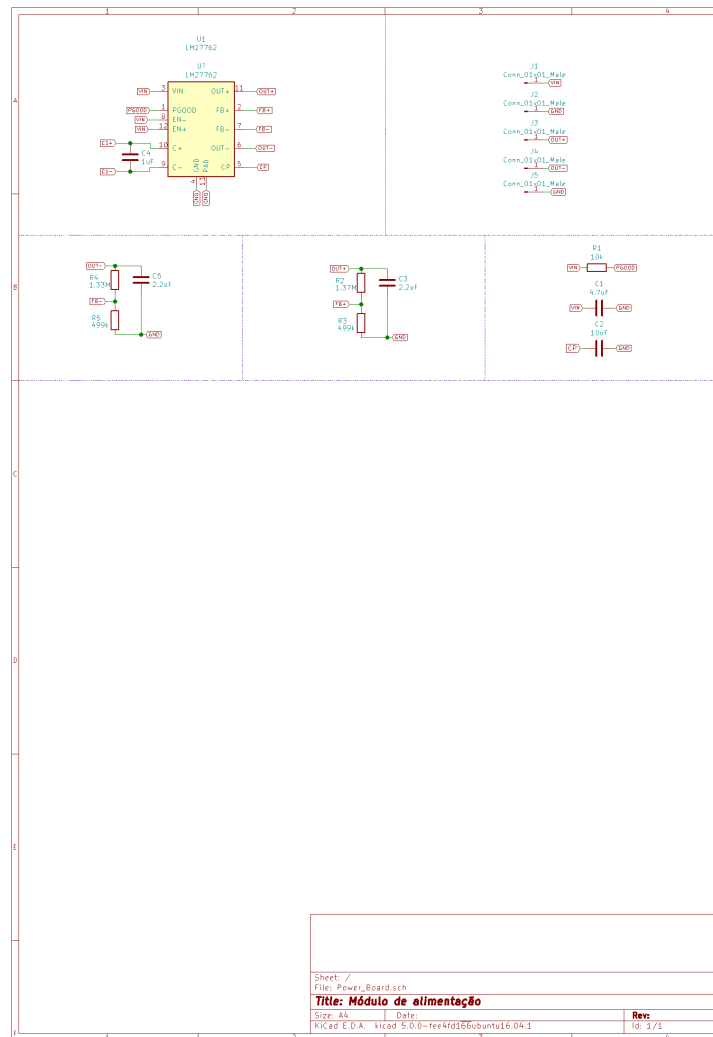


Figura A.1: Módulo de alimentação para o sistema de recepção

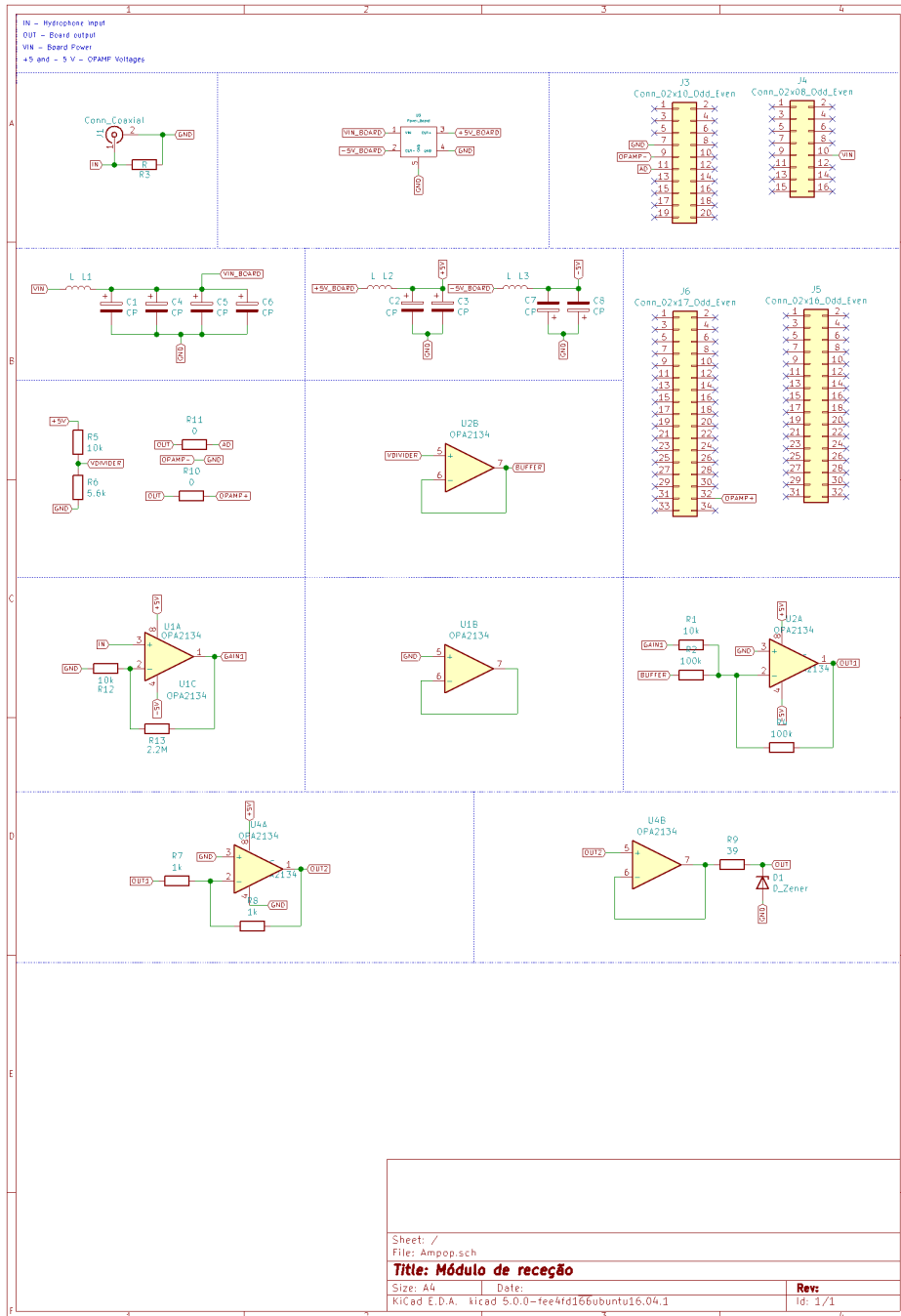


Figura A.2: Módulo de condicionamento de sinal para o sistema de emissão

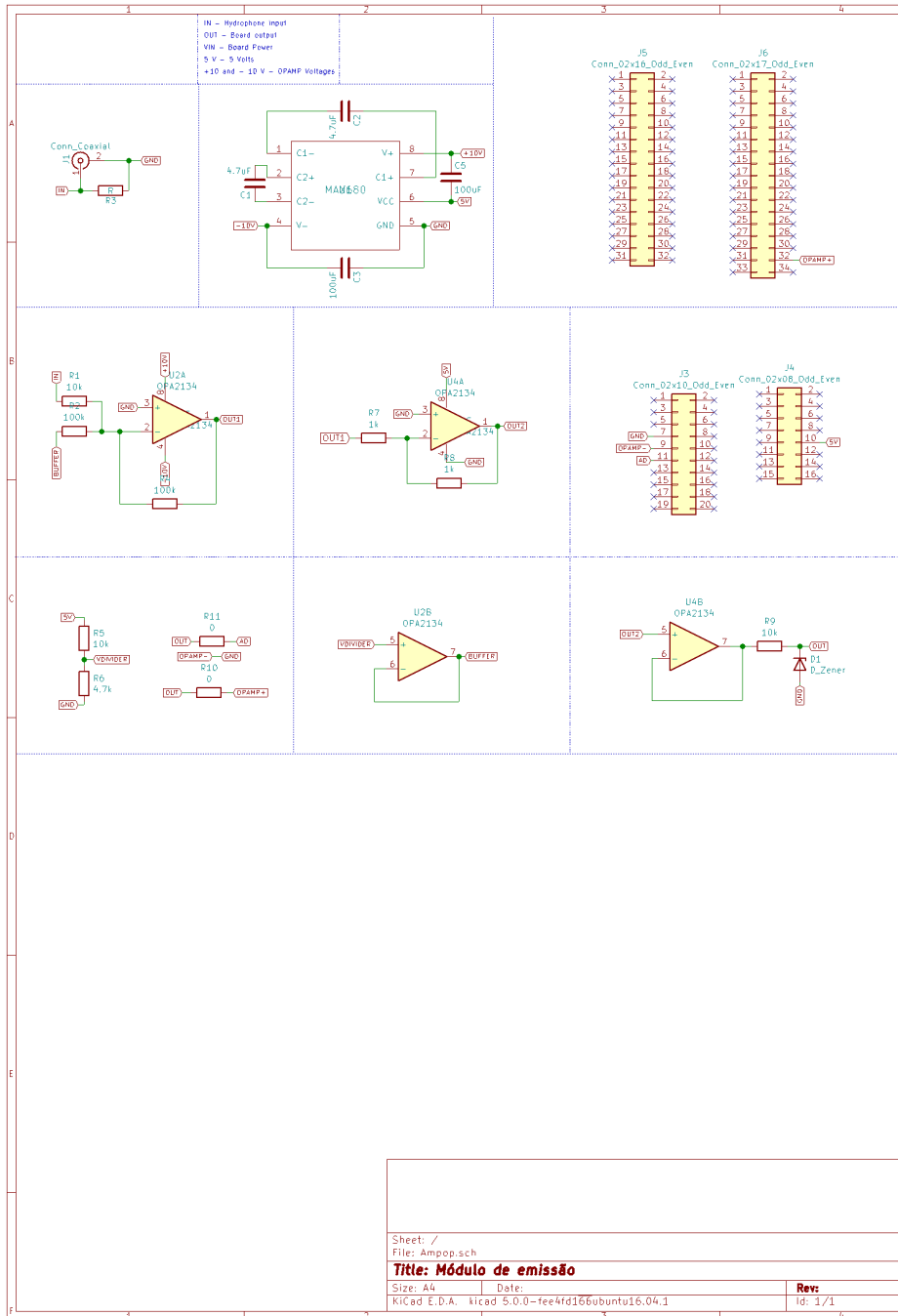


Figura A.3: Módulo de condicionamento de sinal para o sistema de receção

**PÁGINA  
INTENCIONALMENTE  
DEIXADA  
EM BRANCO**

# Apêndice B

## Código

### B.1 Código para a Geração do Sinal

```
1 uint16_t ping[500] = {...}; /* vetor com os valores das amostras de um ping */
2 uint32_t msg[2] = {...}; /* vetor com os intervalos, em amostras, entre pings */
3
4 void TIM4_IRQHandler() {
5     if (TIM_GetITStatus(TIM4, TIM_IT_Update)) {
6         while (1) {
7             /* caso seja para escrever um ping */
8             if (flg_ping == 1) {
9                 /* escrever o valor da sample atual */
10                TM_DAC_SetValue(TMDAC1, ping[i]);
11                i = i + 1;
12                /* caso a escrita do ping tenha chegado ao fim */
13                if (i > size_ping - 1) {
14                    i = 0;
15                    flg_ping = 0;
16                    /* caso seja o final da mensagem, desligar o timer */
17                    if (end == 1) {
18                        TIM_Cmd(TIM4, DISABLE);
19                        TM_DAC_SetValue(TMDAC1, 2000);
20                    }
21                }
22                break;
23            }
24            /* caso seja para criar intervalo entre pings*/
25            if (flg_ping == 0) {
26                TM_DAC_SetValue(TMDAC1, 2000);
27                j = j + 1;
28                /* caso seja altura de voltar a escrever um ping */
29                if (j > msg[h]) {
30                    h = h + 1;
31                    j = 0;
32                    flg_ping = 1;
33                    if (h > size_msg - 1) {
34                        end = 1;
35                    }
36                }
37            }
38        }
39    }
40 }
```

```

37     break;
38     }
39     }
40     TIM_ClearITPendingBit(TIM4, TIM_IT_Update);
41 }
42 }

```

## B.2 Código para a Conversão dos Dados

```

1  /* number: variavel a ser transformada
2  * bit_array: vetor binario
3  * total_bits: numero total de bits alocados para a variavel
4  * integer_bits: numero de bits alocados para a parte inteira
5  */
6
7  void floatToBitArray(float number, uint8_t * bit_array, uint8_t total_bits,
8  uint8_t integer_bits) {
9
10     int16_t i;
11     uint32_t integer_part = number;
12     float decimal_part = fmod(number, 1.0);
13
14     for (i = 0; i != total_bits; i++) {
15         bit_array[i] = 0;
16     }
17
18     /* calculo da parte inteira */
19     for (i = integer_bits; i > 0; i--) {
20
21         if (integer_part % 2 == 1) {
22             bit_array[i - 1] = 1;
23         }
24
25         integer_part = integer_part / 2;
26     }
27
28     /* calculo da parte decimal */
29     for (i = integer_bits; i != total_bits; i++)
30
31     {
32         decimal_part = decimal_part * 2.0 f;
33
34         if (((uint8_t) decimal_part) == 1) {
35             bit_array[i] = 1;
36
37         }
38
39         decimal_part = fmod(decimal_part, 1.0);
40
41     }
42
43 }

```

## Apêndice C

# Testes em Laboratório - Resultados

Tabela C.1: Resultados das posições 1 e 3 para a primeira configuração (34 medições)

	$\Delta_{12}$	$\Delta_{13}$	$\Delta_{id}$
erro médio (amostras)	30	21	17
erro médio (m)	0.09	0.06	0.05
desvio padrão (m)	0.05	0.07	0.05

Tabela C.2: Resultados das posições 2 e 4 para a primeira configuração (34 medições)

	$\Delta_{12}$	$\Delta_{13}$	$\Delta_{id}$
erro médio (amostras)	30	33	18
erro médio (m)	0.09	0.10	0.05
desvio padrão (m)	0.11	0.11	0.06

Tabela C.3: Resultados da posição 5 para a primeira configuração (16 medições)

	$\Delta_{12}$	$\Delta_{13}$	$\Delta_{id}$
erro médio (amostras)	9	34	11
erro médio (m)	0.03	0.10	0.03
desvio padrão (m)	0.02	0.03	0.02

Tabela C.4: Resultados da posição 6 para a primeira configuração (20 medições)

	$\Delta_{12}$	$\Delta_{13}$	$\Delta_{id}$
erro médio (amostras)	13	16	26
erro médio (m)	0.04	0.05	0.08
desvio padrão (m)	0.02	0.10	0.09

Tabela C.5: Resultados da posição 7 para a primeira configuração (18 medições)

	$\Delta_{12}$	$\Delta_{13}$	$\Delta_{id}$
erro médio (amostras)	30	42	10
erro médio (m)	0.09	0.13	0.03
desvio padrão (m)	0.02	0.02	0.02

Tabela C.6: Resultados das posições 1 e 3 para a segunda configuração (28 medições)

	$\Delta_{12}$	$\Delta_{13}$	$\Delta_{id}$
erro médio (amostras)	24	30	19
erro médio (m)	0.07	0.09	0.06
desvio padrão (m)	0.03	0.09	0.05

Tabela C.7: Resultados das posições 2 e 4 para a segunda configuração (31 medições)

	$\Delta_{12}$	$\Delta_{13}$	$\Delta_{id}$
erro médio (amostras)	33	68	21
erro médio (m)	0.10	0.20	0.06
desvio padrão (m)	0.03	0.12	0.06

Tabela C.8: Resultados da posição 5 para a segunda configuração (21 medições)

	$\Delta_{12}$	$\Delta_{13}$	$\Delta_{id}$
erro médio (amostras)	18	47	21
erro médio (m)	0.05	0.14	0.06
desvio padrão (m)	0.07	0.09	0.07

Tabela C.9: Resultados da posição 6 para a segunda configuração (22 medições)

	$\Delta_{12}$	$\Delta_{13}$	$\Delta_{id}$
erro médio (amostras)	9	25	11
erro médio (m)	0.03	0.07	0.03
desvio padrão (m)	0.02	0.02	0.02

Tabela C.10: Resultados da posição 7 para a segunda configuração (25 medições)

	$\Delta_{12}$	$\Delta_{13}$	$\Delta_{id}$
erro médio (amostras)	17	37	10
erro médio (m)	0.05	0.11	0.03
desvio padrão (m)	0.02	0.02	0.02

# Bibliografia

- [1] G. Xu, *GPS Theory, Algorithms and Applications*. Springer Berlin Heidelberg, 2007.
- [2] M. Stojanovic, «Underwater Acoustic Communication», em *Wiley Encyclopedia of Electrical and Electronics Engineering*. American Cancer Society, 2015, pp. 1–12, ISBN: 9780471346081. DOI: 10.1002/047134608X.W5411.pub2.
- [3] N. Viana, P. Guedes, D. Machado, D. Pedrosa, A. Dias, J. Almeida, A. Martins e E. Silva, «Underwater Acoustic Signal Detection and Identification Study for Acoustic Tracking Applications», out. de 2018, pp. 1–7. DOI: 10.1109/OCEANS.2018.8604565.
- [4] K. Vickery, «Acoustic positioning systems. A practical overview of current systems», em *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles (Cat. No.98CH36290)*, ago. de 1998, pp. 5–17. DOI: 10.1109/AUV.1998.744434.
- [5] Y. Han, C. Zheng e D. Sun, «Accurate underwater localization using LBL positioning system», em *OCEANS 2015 - MTS/IEEE Washington*, out. de 2015, pp. 1–4. DOI: 10.23919/OCEANS.2015.7401893.
- [6] A. Tomczak, «Modern Methods of Underwater Positioning Applied in Subsea Mining», *AGH Journals of Mining and Geoengineering*, vol. 35, pp. 381–394, jan. de 2011.
- [7] J. Ferrel e M. Barth, *The global positioning system and inertial navigation*. McGraw-Hill, 1998, p. 340.
- [8] J. Cao, C. Zheng, D. Sun e D. Zhang, «Travel time processing for LBL positioning system», em *2016 IEEE/OES China Ocean Acoustics (COA)*, jan. de 2016, pp. 1–5. DOI: 10.1109/COA.2016.7535752.
- [9] J. J. Leonard, A. A. Bennett, C. M. Smith, H. Jacob e S. Feder, «Autonomous Underwater Vehicle Navigation», jan. de 1998.
- [10] *Kongsberg - High Precision Acoustic Positioning*, visitado em março 2019. [Online]. Disponível em: <http://www.km.kongsberg.com/>.

- [11] Z. Yuyi, G. Zhenbang, W. Lei, Z. Ruiyong e L. Huanxin, «Study of Underwater Positioning Based on Short Baseline Sonar System», em *2009 International Conference on Artificial Intelligence and Computational Intelligence*, vol. 2, nov. de 2009, pp. 343–346. DOI: 10.1109/AICI.2009.83.
- [12] L. Wang e S. Pang, «AUV Navigation Based on Inertial Navigation and Acoustic Positioning Systems», out. de 2018, pp. 1–8. DOI: 10.1109/OCEANS.2018.8604773.
- [13] F. W. Olin, «Integrating Precision Relative Positioning Into JASON / MEDEA ROV Operations», 2006.
- [14] M. Arkhipov, «An approach to using basic three-element arrays in tetrahedral-based USBL systems», em *2013 OCEANS - San Diego*, set. de 2013, pp. 1–8.
- [15] P.-P. J. Beaujean, A. Bon e E. An, «Motion-compensated acoustic positioning in very shallow waters using spread-spectrum ultra-short baseline array processing», jan. de 2008. DOI: 10.1109/OCEANS.2008.5151813.
- [16] M. Arkhipov, «Designing a USBL system based on a square pyramid array with a complete set of three-element arrays», em *2012 Oceans*, out. de 2012, pp. 1–9.
- [17] A. Zielinski e L. Zhou, «Precision Acoustic Navigation for Remotely Operated Vehicles (ROV)», jan. de 2005.
- [18] *High performance USBL positioning system*, visitado em março 2019. [Online]. Disponível em: <http://www.ixblue.com>.
- [19] K. Vickery, «Acoustic positioning systems. New concepts-the future», em *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles (Cat. No.98CH36290)*, ago. de 1998, pp. 103–110.
- [20] J. Stefański, «Hyperbolic Position Location Estimation in the Multipath Propagation Environment», em. ago. de 2009, vol. 308, pp. 232–239. DOI: 10.1007/978-3-642-03841-9\_21.
- [21] W. H. FOY, «Position-Location Solutions by Taylor-Series Estimation», *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-12, n.º 2, pp. 187–194, mar. de 1976, ISSN: 0018-9251. DOI: 10.1109/TAES.1976.308294.
- [22] Y. T. Chan e K. C. Ho, «A simple and efficient estimator for hyperbolic location», *IEEE Transactions on Signal Processing*, vol. 42, n.º 8, pp. 1905–1915, ago. de 1994, ISSN: 1053-587X. DOI: 10.1109/78.301830.
- [23] B. T. Fang, «Simple solutions for hyperbolic and related position fixes», *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, n.º 5, pp. 748–753, set. de 1990, ISSN: 0018-9251. DOI: 10.1109/7.102710.

- [24] J. Abel e J. Smith, «The spherical interpolation method for closed-form passive source localization using range difference measurements», em *ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 12, abr. de 1987, pp. 471–474. DOI: 10.1109/ICASSP.1987.1169674.
- [25] B. Friedlander, «A passive localization algorithm and its accuracy analysis», *IEEE Journal of Oceanic Engineering*, vol. 12, n.º 1, pp. 234–245, jan. de 1987, ISSN: 0364-9059. DOI: 10.1109/JOE.1987.1145216.
- [26] H. Schau e A. Robinson, «Passive Source Localization Employing Intersecting Spherical Surfaces from Time of Arrival Differences», *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 35, pp. 1223–1225, set. de 1987. DOI: 10.1109/TASSP.1987.1165266.
- [27] L. W. Couch, *Digital and Analog Communication Systems*. Pearson, 2013.
- [28] B. Sklar, *Digital Communications*. Prentice Hall.
- [29] S. Haykin, *Communication Systems*. John Wiley & Sons, Inc.
- [30] U. Madhow e M. L. Honig, «MMSE interference suppression for direct-sequence spread-spectrum CDMA», *IEEE Transactions on Communications*, vol. 42, n.º 12, pp. 3178–3188, dez. de 1994, ISSN: 0090-6778. DOI: 10.1109/26.339839.
- [31] Y.-R. Tsai e J.-F. Chang, «Using frequency hopping spread spectrum technique to combat multipath interference in a multiaccessing environment», *IEEE Transactions on Vehicular Technology*, vol. 43, n.º 2, pp. 211–222, mai. de 1994, ISSN: 0018-9545.
- [32] X. Che, I. Wells, G. Dickers, P. Kear e X. Gong, «Re-evaluation of RF electromagnetic communication in underwater sensor networks», *IEEE Communications Magazine*, vol. 48, n.º 12, pp. 143–151, dez. de 2010, ISSN: 0163-6804. DOI: 10.1109/MCOM.2010.5673085.
- [33] N. Farr, A. Bowen, J. Ware, C. Pontbriand e M. Tivey, «An integrated, underwater optical /acoustic communications system», em *OCEANS'10 IEEE SYDNEY*, mai. de 2010, pp. 1–6.
- [34] D. L. Bradley e P. Stern, «Underwater Sound and the Marine Mammal Acoustic Environment A Guide to Fundamental Principles», 2008.
- [35] K. V. MacKenzie, «Nine-term equation for sound speed in the ocean», *Journal of The Acoustical Society of America - J ACOUST SOC AMER*, vol. 70, pp. 807–812, set. de 1981. DOI: 10.1121/1.386920.
- [36] M. Stojanovic e J. Preisig, «Underwater acoustic communication channels: Propagation models and statistical characterization», *IEEE Communications Magazine*, vol. 47, n.º 1, pp. 84–89, jan. de 2009, ISSN: 0163-6804. DOI: 10.1109/MCOM.2009.4752682.

- [37] R. A. Serway e J. W. Jewett, *Physics for Scientists and Engineers*, eighth edi. 2010, p. 1558, ISBN: 9781439048443.
- [38] J. Nitsche e R. Spooner, «Performance estimates of doppler tracking systems», em *Ocean '73 - IEEE International Conference on Engineering in the Ocean Environment*, set. de 1973, pp. 86–91.
- [39] C. Zhongyi, H. Liang Li, P. Lin Liu e Y. Xin Chen, «Navigating Error Analysis for Acoustic Doppler Velocity Log», *Applied Mechanics and Materials*, vol. 303-306, pp. 453–458, fev. de 2013.
- [40] *UAC-2 USB 3.0 Audio Converter*, visitado em abril 2019. [Online]. Disponível em: <http://www.zoom.co.jp>.
- [41] *AS-1 Aquarian Scientific Broadband Measurement Hydrophone*, visitado em abril 2019. [Online]. Disponível em: <http://www.aquarianaudio.com/>.
- [42] J. A. O'Keefe, «The Universal Transverse Mercator Grid and Projection», *The Professional Geographer*, vol. 4, pp. 19–24, fev. de 2005. DOI: 10.1111/j.0033-0124.1952.45\_19.x.
- [43] R. Kalman, «A New Approach To Linear Filtering and Prediction Problems», *Journal of Basic Engineering (ASME)*, vol. 82D, pp. 35–45, jan. de 1960. DOI: 10.1115/1.3662552.
- [44] Y. Bar-Shalom, X. Li e T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.
- [45] *ARM® Cortex®-M4 32b MCU+FPU, 225DMIPS, up to 512kB Flash/128+4KB RAM, USB OTG HS/FS, 17 TIMs, 3 ADCs, 20 comm. interfaces*, visitado em abril 2019. [Online]. Disponível em: <http://www.st.com/>.
- [46] A. S. Sedra e K. Smith, *Microelectronic Circuits*, seventh edition. 2015, p. 1472.
- [47] *OPAx192 36-V, Precision, Rail-to-Rail Input/Output, Low Offset Voltage, Low Input Bias Current Op Amp with e-trim*, visitado em abril 2019. [Online]. Disponível em: <http://www.ti.com/>.
- [48] *Maxim +5V to +10V Converters*, visitado em abril 2019. [Online]. Disponível em: <http://pdfserv.maximintegrated.com/>.
- [49] *50 kHz High-Pass Filter*, visitado em abril 2019. [Online]. Disponível em: <http://www.thorlabs.com/>.
- [50] *UM1974 User manual*, visitado em abril 2019. [Online]. Disponível em: <http://www.st.com/>.
- [51] *32-bit Arm® Cortex® -M7 480MHz MCUs, up to 2MB Flash, up to 1MB RAM, 46 com. and analog interfaces*, visitado em abril 2019. [Online]. Disponível em: <http://www.st.com/>.

- [52] *LM27762 Low-Noise Positive and Negative Output Integrated Charge Pump Plus LDO*, visitado em abril 2019. [Online]. Disponível em: <http://www.ti.com/>.
- [53] V. Meireles, *Circuitos Eléctricos*. Lidel, 2009.
- [54] M. Quigley, B. Gerkey e W. D. Smart, *Programming Robots with ROS*. O'Reilly Media, 2015.