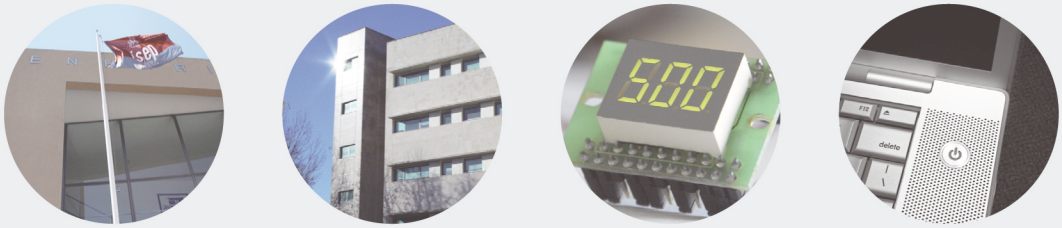
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES



Bearing Based Low Cost Underwater Acoustic Positioning System

PEDRO EMANUEL DE ALVES GUEDES
Outubro de 2019

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Bearing Based Low Cost Underwater Acoustic Positioning System

Pedro Emanuel de Alves Guedes

Master's Degree in Electronics and Computers Engineering

Supervisor: Alfredo Manuel de Oliveira Martins

Co-Supervisor : José Miguel Soares de Almeida

October 8, 2019

© Pedro Emanuel de Alves Guedes, 2019

“ Por este rio acima
Os barcos vão pintados
De muitas pinturas
Descrevem varandas
E os cabelos de Inês
Desenham memórias
Ao longo da água”

Fausto Bordalo Dias, Por este Rio Acima

THIS PAGE
INTENTIONALLY
LEFT BLANK

Abstract

The Ocean Robotics turned into one of the major fields of research since the exploration of oceans brings many benefits to the human condition. Remotely operated vehicles (ROVs) and Autonomous Surface Vehicles (ASVs) are the common instruments used in this medium, since they prevent human losses and enable more and reliable data for the projects they are inserted in. Most scenarios include support vehicles such as ASVs for monitoring purposes since they are able to use specialised positioning systems such as Global Positioning System (GPS), which are ineffective in underwater environments. This is due to electromagnetic signals used by GPS being attenuated by the medium. As an alternative, acoustic solutions are used. Underwater Acoustic Positioning Systems (UAPSs) have always been an important field of study being used in multiple marine applications. Acoustic fish tracking allows for behavioural and *in-situ* fish population studies. This process usually involves tagging fishes with acoustic emitters (i.e. tags) and the usage of acoustic receivers. Robotic autonomous vehicles can then be used to carry the acoustic receivers in order to dynamically cover a greater mission area, improving the efficiency of the localisation of acoustic sources.

An acoustic tag detector was developed to have real-time detection and identification of acoustic signals. A Direction of Arrival (DoA) algorithm was developed from ground up to enable tracking applications. This dissertation presents the improved results of this new system as well as the tests that were made to the DoA algorithm in a simulated environment. Additionally, the position estimation is improved using a Kalman Filter. This work was developed in the context of the MYTAG Portuguese R&D project, addressing the study and characterisation of European flounder migrations and to be applied to any target that has a known acoustic signals. One of the objectives of this project is to eventually use an ASV to track a set of flounders, namely with the ROAZ ASV. The use of an unmanned surface vehicle allows for a non-static baseline. In the proposed solution the acoustic signals are tracked with a system composed of three acoustic receivers that are linked to the same computer using a synchronised time source. Not only that but this solution provides two possible methods for the main objective which is to track targets. These methods enable the possibility to estimate the target's position in the world, while developing a low-cost solution with a newly developed DoA algorithm.

THIS PAGE
INTENTIONALLY
LEFT BLANK

Acknowledgements

This work is dedicated to all who supported me in this constant struggle that was the culmination of my masters degree.

I would like to thank my supervisor, Prof. Alfredo Martins, and my co-supervisor, Prof. José Almeida, for all the advice they have given me and for the way they have always encouraged me to strive for more and better.

I thank my parents and sister for their eternal patience in those less good times that had arisen, but also for all the strength they gave me when even I thought it would be impossible to find it.

I thank all the friends that accompanied me to *Laboratório de Sistemas Autómos* and all the new ones that I have made there. All of them were always there to cheer me up and to help me when no solution was any near to my grasp.

I would like to express my thanks to the rest of my family who supported me at all times in my life, praying for me and my success without ceasing.

And I can never forget all the friends I made during this journey and those who have always been here. Their voice and words of strength have always been with me all the way and I will never forget that. Not even those who went several kilometres away from here, but never from my mind.

Thank you all,

Pedro Guedes

THIS PAGE
INTENTIONALLY
LEFT BLANK

Contents

List of Figures

THIS PAGE
INTENTIONALLY
LEFT BLANK

List of Tables

THIS PAGE
INTENTIONALLY
LEFT BLANK

List of Abbreviations

2D	**2**-**D**imensional
3D	**3**-**D**imensional
ADC	**A**nalog-to-**D**igital **C**onverter
AoA	**A**ngle **o**f **A**rrival
ARM	**A**dvanced **R**ISC **M**achine
ASV	**A**utonomous **S**urface **V**ehicle
AUV	**A**utonomous **U**nderwater **V**ehicle
BNC	**B**ayonet **N**eill**C**oncelman
CFE	***C**entro de **E**cologia **F**uncional*
CIIMAR	***C**entro **I**nterdisciplinar de **I**nvestigação **M**arinha e **A**mbiental*
DAC	**D**igital-to-**A**nalog **C**onverter
DMA	**D**irect **M**emory **A**ccess
DoA	**D**irection **o**f **A**rrival
DOAE	**D**irection **o**f **A**rrival **E**stimation
DGPS	**D**ifferential **G**lobal **P**ositioning **S**ystem
GIB	**G**PS **I**ntelligent **B**uoys
GPS	**G**lobal **P**ositioning **S**ystem
IC	**I**ntegrated **C**ircuit
ID	**Id**entification
INESC TEC	***In**stituto de **E**ngenharia de **S**istemas e **C**omputadores, **Te**cnologia e **C**iência*
ISEP	***I**nstituto **S**uperior de **E**ngenharia do **P**orto*
KF	**K**alman **F**ilter
LSA	***L**aboratório de **S**istemas **A**utónomos*
lwIP	light**w**eight **IP**
MARE	**Mar**ine and **E**nvironmental Sciences Centre
MEEC	**M**estrado **E**ngenharia **E**letrotécnica e de **C**omputadores
LDO	**L**ow-**D**ropo**o**ut Regulator
Opamp	**Op**erational **Amp**lifier
PE	**P**osition **E**stimation
PCB	**P**rinted **C**ircuit **B**oard

PWM	**P**ulse **W**idth **M**odulation
RF	**R**adio **F**requency
RMS	**R**oot **M**ean **S**quare
ROS	**R**obot **O**perating **S**ystem
ROV	**R**emotely **O**perated **V**ehicles
RTT	**R**ound-**T**rip **T**ime
SSBL	**S**uper **S**hort **B**aseline
SVP	**S**ound **V**elocity **P**rofile
TDoA	**T**ime-**D**ifference-**o**f-**A**rrival
ToA	**T**ime **o**f **A**rrival
UAPS	**U**nderwater **A**coustic **P**ositioning **S**ystem
uC	**MicroC**ontroller
UDP	**U**ser **D**atagram **P**rotocol
USB	**U**niversal **S**erial **B**us
VPS	**V**emco **P**ositioning **S**ystem
VRAP	**V**EMCO **R**adio-**A**coustic **P**oisitoning
VRU	**V**ertical **R**eference **U**nit

Chapter 1

Introduction

Currently there is more information about Mars' surface than Earth's. That's justified by the enormous amount of water that covers this planet, in fact, almost two thirds of it [1], being the reason why it's called the Blue Planet. In the past few years, the scientific community has been making efforts to develop new technologies which allow to explore and study the oceans. Many marine species are yet to be discovered and classified, as well as minerals at the bottom of the oceans [2]. Nowadays the scientific community has turned to Ocean Robotics in search of solving problems inherent to this environment that usually are easy to solve and taken for granted on the surface, highlighting, for example, the real-time localisation. Positioning a target is an important daily necessity, for example when using the Global Positioning System (GPS) to know the current location and which path to take depending on the destination. This is even more important in robotics, as it raises the possibility of generating several applications, highlighting the context of this dissertation: tracking a target. However, it is not possible to use technologies such as GPS in the underwater environment, as the medium attenuates the signals used by this solution, preventing their propagation. Acoustic signals are generally used for communications in this medium.

The localisation of acoustic signals allows to:

- Detect underwater autonomous vehicles.
- Monitor underwater species with acoustic emitters.
- Create tracking applications.
- Decode messages sent by acoustic transducers.

This research was carried out within the scope of Electronics and Computers Master's (MEEC) degree Thesis/Dissertation (TEDI), a subject of *Instituto Superior de Engenharia do Porto* (ISEP). The project allowed to respond to *Laboratório de Sistemas Autónomos* (LSA) needs and was done in partnership with *Instituto de Engenharia de Sistemas e*

Figure 1.1: Roaz ASV

Computadores, Tecnologia e Ciência (INESC TEC), aiming to solve a current issue: study and characterise the European flounder's (*Platichthys flesus*) migrations which is the main requirement of the MYTAG project. The MYTAG project proposes the use of artificial acoustic emitters, called tags, and the development and application of new technological solutions to detect, identify and track the acoustic signals sent by the emitters while using an autonomous surface vehicle (ASV) [3], such as ROAZ in Fig 1.1. MYTAG served as a foundation to develop an acoustic target tracking system.

Since it was an ongoing project and already under study in [3], there was a need to establish whether the resources previously used required to be improved or to be developed from scratch.

1.1 Objectives

The main objective to be fulfilled was to validate the concept of creating an acoustic tracking system that could be used in the MYTAG project or in other applications with similar requirements. This goal can be divided into several sub-goals listed below:

- Using the MYTAG project as a mean to create an acoustic target tracker
- Design and validate new hardware if needed.
- Develop and validate software to acquire and process data from the underwater domain.
- Develop software for tracking the V7 tags after the above was accomplished.
- Validate the whole solution.

1.2 Thesis Structure

Chapter 2 describes and formulates the thesis problem and what was the state of the MYTAG project.

Chapter 3 describes the state of the art of the technology used nowadays to answer to the problems this thesis present.

Chapter 4 is an exposition of the underwater communications technologies and the sources of errors of this medium.

Chapter 5 summarises how the problem was addressed and the methods used to solve it. Basically it is responsible for making the link between theory and the developed work.

Chapter 6 reprises the acoustic signal detection and identification explaining how those algorithms function and how they are crucial for the solution.

Chapter 7 presents the developed Direction of Arrival algorithm as well as the synchronisation Algorithm.

Chapter 8 describes how the position is estimated through the use of a developed from scratch simulator as well as the implementation of a Kalman Filter.

In Chapter 9 both the hardware and software of the solution are explained and validated.

Chapter 10 has the solution's results.

In Chapter 11 a conclusion is made regarding if the objectives mentioned in Chapter 1 were accomplished and possible ideas for future work.

THIS PAGE
INTENTIONALLY
LEFT BLANK

Chapter 2

Problem Formulation

This chapter arises from the need to clarify and contextualise the problem to be solved that was mentioned in the introductory chapter of the dissertation, which is the development of an acoustic tracking system in the underwater environment. As said the need to develop this system is due to the requirements of the MYTAG project.

The MYTAG project is a Portuguese R&D project, addressing the study and characterisation of the European flounder (*Platichthys flesus*) migrations in the northern estuarine environments of Portugal. The flounder, taken as a model species, migrates along the river-estuary-sea during its lifetime and with high variability in these migrations throughout its distribution zone. The biologists aim to understand the links between the various flounder developmental states and to reveal the plasticity of their migratory strategies. This multidisciplinary research project integrates the use of natural and artificial tags for tracking the fish populations.

MYTAG pretends to implement a multi-tag approach applicable to as many species worldwide, enabling, in addition, the development of new technology. The MYTAG project is composed by a team of professionals from *Centro de Ecologia Funcional* (CFE), *Centro Interdisciplinar de Investigação Marinha e Ambiental* (CIIMAR), *Centro de Ciências do Mar e do Ambiente* (MARE) and from INESC TEC. The used acoustic emitters chosen by the biologists' team are the V7 tags from VEMCO, surgically inserted in the flounders, as Figure 2.1 shows. The V7 tag was chosen primarily due to size restrictions since biologists intended to also study the behaviour of young fish populations with limited size. These tags operate at 69 kHz [4] and emit a codded identification with a frequency of emission ranging between 90 s and 120 s. These tags are only capable of pinging a signal without the capability of processing or receiving data from other acoustic modems, since their only purpose is to be identified. Not having any sensors and long intervals between emissions allows the tags to have a longer battery life, lasting for approximately 255 days [4]. Since the acoustic tracking system was based on solving technological problems related to the MYTAG project, the V7 tags turned out to be the target to follow.

Figure 2.1: Tag surgically inserted in a flounder

Initially two receiver solutions were considered: a receiver made by VEMCO, the VR2W, and a solution used in [3].

The VR2W receiver is only capable of receiving signals codified at 69 kHz and was made to identify the VEMCO acoustic emitters, not being able to be used outside this scope. Those receivers were able to detect the used tags, however the main objective of this project is to not only detect and identify the signals but also to track and/or position the tags/flounders in the world. VEMCO has the VPS service, which is mentioned in Chapter 3, that can log the tag's position if three receivers are positioned as they advise. With those conditions no real-time data could be attained nor be implemented in a tracking solution with a robot. More tests were made and if tags send their signals simultaneously or were at a distance bigger than 100 m, as shown in the results of the published paper [3], they won't be identified.

The use of robotic autonomous vehicles can provide useful area coverage and integrated information gathering thus providing added efficiency to the process.

For the requirements of the project to be met a system that was able to identify and

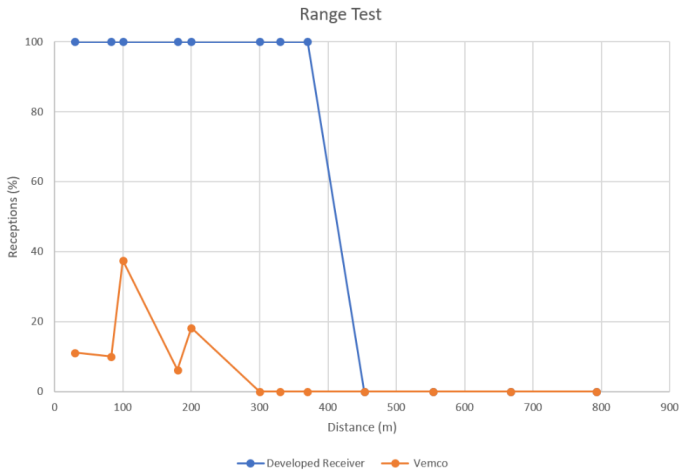


Figure 2.2: VEMCO tests compared to a previously developed solution [3]

detect the V7 tags' signal needed to be developed. Not only that but near to real-time data needed to be attained so it could be used by the ASV responsible for tracking the targets, in this case ROAZ. However, for that to be possible more data was needed besides the tag's identification (ID). The hardware and software used in [3] needed to be improved to answer the project's requirements. This assumption arises since that solution used a ZOOM UAC-2 soundboard to record the signals, with a sampling rate of 192 kHz, which was barely above the *Nyquist* frequency. Not only that but the recordings were made with an Open-source software and not directly recorded to the computer, which could generate more delays.

THIS PAGE
INTENTIONALLY
LEFT BLANK

Chapter 3

State of the Art

This chapter addresses current technologies and techniques used for Underwater Acoustic Positioning Systems (UAPSs), as well as solutions available in the market. This study allows to identify techniques that can be relevant to the thesis' problem. A relationship is also made between the studied state of the art and the main system requirement, describing how the proposed solution should be approached.

3.1 Underwater Acoustic Positioning System

In recent years, Ocean Robotics turned into one of the major fields of research [5], since the oceans' exploration brings many benefits to the human condition, such as underwater work, mineral and oil exploration, ocean sciences, salvage operations and many others [6]. Robotic systems, Remotely Operated Vehicle (ROV), ASV, Autonomous underwater vehicle (AUV), are used in ocean operations mainly to tackle with the challenges of the environment, such as the depth, large areas and hazardous environment conditions reducing human risk. In many cases (such as deep sea operation) are the only tool possible. To ensure robots can "survive" their missions in the underwater environment, their position must be known, among other parameters. Tracking and localising applications are useful for that purpose. If an ASV is used as a support vehicle it will be above the water and it is able to attain its position through GPS and other systems. The same can't be said in the underwater environment, since the electromagnetic signals used in GPS are attenuated by the medium [7], [8], making the GPS information unavailable. The UAPSs were developed to detect and/or track targets in those missions. The connection between the systems above and underwater grant complement information to the already implemented navigation system in the AUVs [5], offering absolute positioning and complementary information.

Most of UAPSs use acoustic receivers (hydrophones) usually paired with acoustic emitters. Transponders are used in most of the applications. Those are devices that are capable

of both emitting and receiving signals, even if they use the same circuitry for those purposes. Many hydrophones are capable of doing so. The typical measurements attained from the UAPSs are the time-difference-of-arrival (TDoA), the time-of-arrival (ToA) and the angle-of-arrival (AoA) or Direction of Arrival (DoA) [8]. Classical approaches , such as the Long Baseline (LBL), Short Baseline (SBL), and Ultra-Short Baseline (USBL) systems, use those types of measurements to localise an acoustic source [6]. Those approaches occupy most of the market.

3.2 Solutions available on the market

This section summarises the typical solutions that can be found currently on the market. Classical solutions like the LBL, SBL and USBL, along with more recent solutions that are mostly adaptations of the latter are addressed.

3.2.1 LBL

The LBL is a method that is used to determine the position of an acoustic emitter [6]. That is achieved by using seabed beacons and a transceiver installed on a target [9]. This method's name derives from the fact that the beacons that compose the baseline are spaced within large distances, in the order of 50 - 2000 m [10], when compared to the other approaches mentioned in this section [11]. The beacons in LBL can be compared to the satellites used in GPS [9]. By using seabed beacons, there is no need to compute the reference frame changes [11]. The target sends an acoustic signal, being detected by the beacons, which will send a response, having a configuration similar to what is seen in Figure 3.1, where dn refers to the range between the transponders (Tn) and the ASV

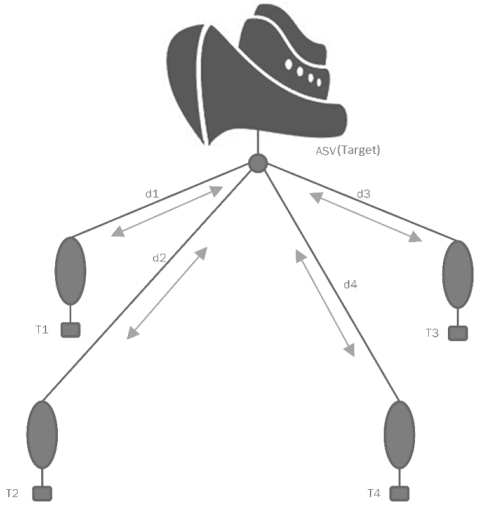


Figure 3.1: LBL configuration

(target). The time interval between the first emission and the response is the raw data used to compute the distance between the target and each beacon [10], [11]. Knowing this, the target's localisation can be computed after two steps: through trilateration, using the distances to the beacons, resulting in relative position in a coordinate frame associated to the beacons. Secondly, a change to geodetic coordinates is made, with the absolute positions of the beacons and the target [9], [12]. The range measurements are often complemented with the use of depth sensors, to easily determine depth of the target [6]. The quality of the computed position is directly affected by the time interval measurement between target and beacons, as well as the calibrated positions of the seabed beacon array [9], both relative to each other and absolute [12]. LBL requires, as the GPS system, at least 3 receivers, wherein a fourth will provide a quality check and redundancy to the measurements [10].

This method is considered as the most accurate [13] and it is commonly used in underwater construction, mining operations and many engineering/scientific fields [13]. However, it is not free of inconveniences: it has an area of operation limited to the reference beacons [14], the need of two-way of travel ranging and the necessity to calibrate the positions of the seabed beacons array, in addition to deploy them on the seabed [15]. Placing and recovering the beacons from the seabed is made with professional divers, which will introduce errors to the beacons array as well as an increase in the price [14]. There are some examples of LBL in the market such as SM-975/976 from Teledyne and the SIMRAD MPT 319 from Kongsberg.

3.2.2 SBL

The SBL systems are often used on tracking applications, rescue operations and where targets are found in low depths [14]. These systems avoid the constraint of having a fixed area of operation, since there is no need to place the transponders in the seafloor, setting them, instead, on the bottom of a surface vessel or at its sides. One transceiver is used on the target and at least three beacons are placed in the vessel [6]. The name derives from how the baseline is placed, in other words, the transceivers or transponders are close to one another [16], guaranteeing that the distance between each other is less than the one to the target(s) [11].

The SBL system allows the DoA and/or range computation of the array of transceivers in the vessel relatively to target [14]. As the LBL, time measurements are the raw data in this system [14], being the signal round-trip time (RTT) or ToA, depending on the configuration [17]. Knowing this, the system can function in two ways:

- Transponder mode (flight interrogation technique): RTT measurements are made, where one acoustic signal is sent by one of the vessel's transponders, being then received by the target's transponder, which will "answer" back to the vessel, resulting

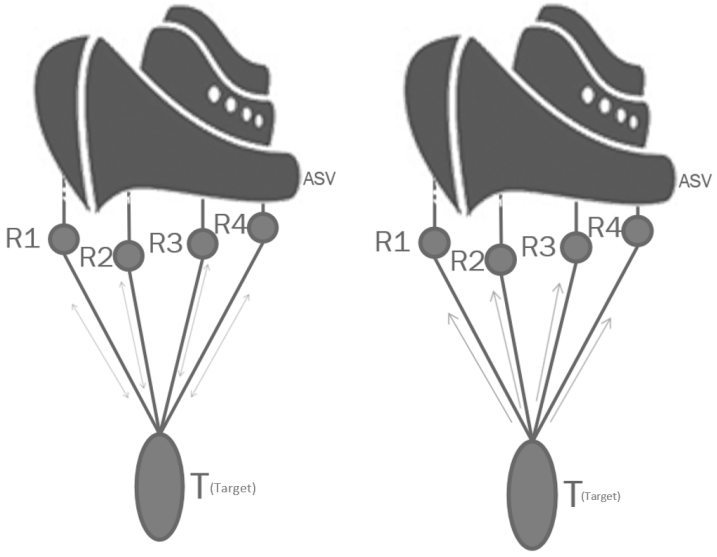
in range and direction measurements between each transponder in the vessel and the target [18], [19]. This method can be seen in Figure 3.2a, where Rn are the vessel's transponders and T the target.

- Pinger mode: TOA measurements are made, where the target sends an acoustic signal, being detected afterwards by the each transceiver in the vessel. With those measurements a DoA of the acoustic source can be computed [14], through the TDoA. This is illustrated in Figure 3.2b, where Rn are the vessel's transponders and T the target.

The computed bearing and/or range from this system is relative to the location of the baseline [14], [16]. Knowing this and that the vessel might move, it's advisable that the system is also composed of a GPS receiver, for the vessel and a Vertical Reference Unit (VRU) and Gyro, for the baseline, in order to provide a final position that is earth referenced [14]. In addition, pitch and roll measurements can also be made to compensate the curling [14].

Performance wise the SBL system will have better measurements the higher it is the spacing between the baseline's transponders [16]. With that said, where space allows, such as large vessels, the SBL will have a precision similar to LBL [14], [16]. This method, however, needs a large baseline when operating at big depths, above 40 m, precise baseline calibration and extra sensors for having the earth-referenced positions of the acoustic source [14], [15].

In the market it is possible to find solutions from manufacturers such as KONGSBERG's, Sonardyne and Nautronix.



(a) SBL in Transponder mode (b) SBL in Pinger mode

Figure 3.2: SBL configurations

3.2.3 USBL

The USBL is a method commonly used for target tracking with a support vessel [20]. It is also called Super Short Baseline (SSBL) in some literature. Currently, using the USBL as a tool for AUV navigation is not yet a common practice, but studies have been made to make it a possible solution [20]. The USBL is similar to the SBL, since it uses an array of at least three transponder elements, located in the horizontal plane, usually on the vessel's bottom. The data's reliability and accuracy can improve if the array of transponder elements in the vessel is increased [21], [22].

Usually this method assures that a target has a transponder, which allows the reception and emission of acoustic signals [20], [21]. It relies on a baseline of 10 cm or less [21] between each vessel's transponder. Since those are the typical size of a baseline, usually, the sensors are built into an assembly in close proximity, resembling only one sensor [15]. This guarantees a more compatible and easy to deploy solution, when compared to LBL (Subsection 3.2.1). As the other methods mentioned in Subsections 3.2.1 and 3.2.2, time measurements are the raw data which allow the computation of a position [23]. An acoustic signal is sent by the vessel's transponders being then detected by the target's transponder, which replies with its own signal. This reply is then detected by the vessel's setup [21]–[23].

The location of underwater objects is determined by computing the angular position and/or slant range of the system to the target [22] in the reference coordinate system determined by the transponder array in the vessel [23]. USBL position and range estimation is mainly based on bearing angles estimation of the acoustic sources that travel between the vessel and the target, where precise bearing results depend on phase shift measurement of both ends [14], [23]. The slant range is determined by measuring how long the acoustic emission and reception takes. The angular position of the target is calculated by processing how long the acoustic signal reply took to reach the vessel's transponder [22], making the assumption that the wavefront of the acoustic signal is planar when it's received at the vessel's transponder array [23].

To convert the position of the target in geodetic coordinates, extra sensors are needed, such as a gyrocompass, attitude sensor and GPS or Differential Global Positioning System (DGPS). Combining the data from those sensors with the computed position, alongside with coordinate transformation, the absolute position is known [14], [21], [23]. The results can be transmitted to the target, to improve, for instance, its navigation if it is an AUV [20]. There is also an USBL inverted configuration, where the processing is made in the target, allowing it to track the support vessel and easily dock [21], [23].

As it is, the USBL has low system complexity making it an easy tool, avoiding the need to deploy sensors on the seafloor. Not only that, but only "one" sensor is used in the surface vessel [14]. It has better performance within short range or in shallow water [21].

Only one calibration needs to be made during the installation and the accuracy of the system depends on it. The accuracy varies with the range and with the baseline, as well as the vessel self-motions [14], [21], [23]. The system needs to take into account that more errors are added because of the extra sensors needed to have the geodetic coordinates. As in any of the methods errors will be found concerning time measurements [14], [21], [23].

There are some commercial solutions, such as the HiPAP Family from Kongsberg, Micro Nav from TRITECH, Trackit USBL System from Teledyne and Easytrak USBL Systems from Applied Acoustics.

3.2.4 GPS Intelligent Buoys (GIB)

As an alternative concept to the LBL the GIB system has been implemented and it's commercially available. This type of system consists in buoys equipped with GPS receivers and submerged hydrophones, responsible of tracking the position of an underwater target that also has an acoustic emitter. As the methods previously mentioned in Subsections 3.2.1, 3.2.2 and 3.2.3, the raw measurement of the system is time, more specifically, the ToA of the acoustic signals when the signal sent by emitter is received by the surface buoys [5], [24], [25].

Usually, four surface buoys are used and equipped with DGPS receivers and submerged hydrophones. The detected ToA is recorded in an on-board underwater platform. A central station receives the buoys' data via radio, containing the position of the desired underwater target [5], [24], [25]. The depth of the target is available, because it has a depth sensor and its data is coded into the acoustic signal that is sent [5], [24], [25]. ToA values allow the computation of ranges between the buoys and the target, assuming that the sound of water velocity is known. The biggest source of error is from multipathing [5], [24], [25].

The GIB system configuration allows the computation of accurate results similar to the LBL system, avoiding, in addition, the sensors' placement in the seafloor, as well as their calibration. Typically various arrays of GIBS are deployed to extend the mission's area of operation. [25]. This system is used mostly for tracking purposes.

An example of a commercial solution of this system is the GIB SAR, from ALSEAMAR.

3.2.5 Vemco Positioning System (VPS)

The VPS is a method which uses three receivers and relies on a TDoA algorithm used by the VEMCO Radio-Acoustic Positioning (VRAP) [26], [27]. The VPS is mainly used in marine studies, estimating non-real-time underwater acoustic fine-scale positions on multiple tagged animals [26], [27].

The system has receivers and transmitters deployed by the user in the mission area, resembling something as Figure 3.3 represents. The data can be extracted from the receivers

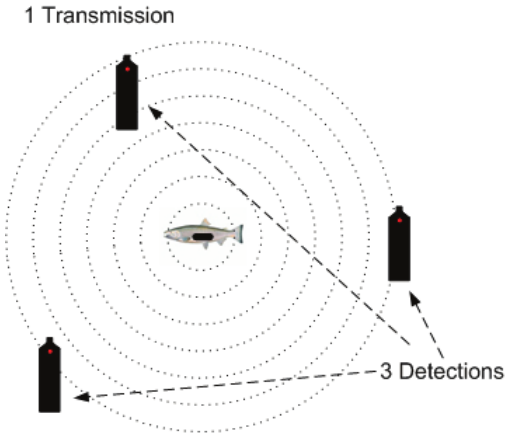


Figure 3.3: VPS [26]

to be then processed by a VEMCO's processing service [26]. The receiver's positions can only be known if the user provides GPS coordinates. The layout of the receivers is discussed with VEMCO. Depending on the results, VEMCO can recommend changes in the layout's design to improve performance [26]. The receivers, however, do not have clocks synchronised to real time [27]. To solve the latter, VEMCO provides Synchronisation tags to correct the clock's drift [26].

For the VPS to work , at least 3 VEMCO receivers need to detect a tag's transmission. The user must guarantee that the receivers are placed close enough, for better performance. Knowing this, to ensure that the a mission area is completely covered, many receivers should be deployed [26].

The system can ensure GPS precision in the localised positions [26]. However, for that to be possible the VEMCO's requirements must be met, the data can't be processed in real-time and both the receivers and transmitters must be bought to this brand. In addition to that many receivers must be acquired, to assure that the mission are is covered and the performance is good enough to have good data.

3.3 Underwater positioning algorithms

In this section several methods are going to be summarised regarding how the position of a target can be estimated and what are the outputs of each method and with which conditions they can be achieved. A target's position can be found through various techniques. Some examples are the hyperbolic positioning, trilateration, triangulation and even through estimation, if other measurements are computed, for example, the direction of the acoustic source.

For the purpose of this thesis mainly two methods of localising a target were studied: hyperbolic localisation and direction of arrival estimation (DOAE)/DoA. This decision is

justifiable, since the project must rely on the minimum number of receivers to make it a low-cost solution when compared to the ones available in the market and because it must be implemented in an ASV. In addition, the V7 tags don't have any depth sensors that can be used to determine, more easily, the 3D position, and are only capable of pinging a signal.

Both the methods depend on the computation of the arrival times of a signal in n pairs of receivers, as the literature suggests [28], [29]. With that said, the next paragraphs are going to expose the most relevant methods.

For both DOAE and hyperbolic localisation a distinction needs to be made: if they are 2-dimensional (2D) or 3-dimensional (3D), which means if the sound source is to be localised in a plane or in a full 3D-space [28], [29].

For the hyperbolic localisation, the TDoA is measured. Knowing the velocity of the sound in the medium, a distance interval between the receiver and transmitter can be computed, through the time differences. However, the position can only be computed through hyperbolic equations, which are non-linear [30], [31]. Because of this, many studies were made to simplify the way the non-linear equations can be solved, either through mathematics or with the used hardware, which can facilitate the problem [31]. Many methods were developed to solve this issue in two dimensions, like Y. T. Chan's in [28], [30]–[34], where not only the position is found through a deduced equation, but also improves the precision when more receivers are available, in other words, more TDoA measurements. However, this method requires a *priori* knowledge of the approximation of the target's location and the distance between it and the receivers [31]. Also in [34], the Foy's method linearises the hyperbolic set of equations by using a Taylor-series expansion. An iterative method is then used to solve the set of linear equations that results from the expansion. As with Chan's method, the iterations start with an initial guess which can improve at each iteration [31], [34]. But there is no way to know if at each iteration the results that are estimated are closer to reality or stuck at a false positive [31]. Some methods are hybrid, relying on both TDoA and TOA measurments, like the one found in [35] made by Yaro, which applies a passive multilateration algorithm using at least 4 receivers.

The angular location of a sound source is estimated by computing the pair-wise time delays of a pair of hydrophones, which can be performed by cross-correlating the recorded signals with known signals [30], [31]. The interval where the cross-correlation has its maximum is taken as the TDoA between the signals [30]. The DOAE only determines the direction of the acoustic source and not the range between the source and receiver [28]. In [28] a study was made for the DOAE and localisation with acoustic sensors. The study distinguishes the 2D and 3D problems with different solutions. The far-field and near-field models are addressed and how the equations for each vary. Not only that, but also how it is possible to estimate directions in 3D with a minimal number of receivers, depending

on the situation. However most of the solutions present in [28] depend on the far-field model, which isn't always applicable, and when the near-field model is used the number of receivers increases, alongside with the complexity of the equations and configurations used [28], [29].

3.4 State of the art discussion

While searching for the state of the art of current technologies used for UAPSs, a variety of solutions was found, both for commercialised systems as well as algorithms and mathematical methods for Position Estimation (PE). Many of the methods assume that the target can receive and send messages, which is not what is in study for the purpose of this thesis. Not only that, but also the assumption that the system has access to the target's depth. Most of the systems that were addressed had specific solution for specific configurations, not being applicable for most of the situations. An increased number of hydrophones to reduce the number of equations needed for each unknown variable is also used. Since the system must be implemented in an ASV and the target won't do more than pinging a signal, a solution that is based in SBL and DOAE/DoA must be implemented. The main goal is to detect the bearing of the acoustic signal relative to the boat, without knowing the range to the target, therefore an hybrid solution must be implemented, guaranteeing that the minimum number of hydrophones is used (low cost solution).

THIS PAGE
INTENTIONALLY
LEFT BLANK

Chapter 4

Underwater acoustic communications

With the rising need of underwater exploration more technologies have been developed to obtain environment data and to provide communication of what is gathered or other relevant information. Not only, that but most applications require for the signals to be received and interpreted in real-time.

Many technologies exist to allow exchange of information in this environment even if acoustic signals are used more often [36]. The underwater communication can be established using acoustic, radio frequency (RF) and optical waves [37], [38]. Even though optical waves have a high data rate and low latency, while being power efficient [37], these signals suffer from high attenuation and scattering in this medium [39]. To avoid or minimise those issues, technologies that use optical waves usually are limited to short distances and predictable situations, such as knowing the initial whereabouts of a target [39], since the line-of-sight is needed. On the other hand, the RF waves can be transmitted using frequencies that vary between 3 kHz and 30 kHz [37], [40] and to have good results, because of the high attenuation, high power antennas are needed [40]. As in [40] Table 4.1 summarises the pros and cons of each technology.

Many messages can be sent through the underwater medium which can contain various types of data such as images, telemetry and monitoring/control[41].

As mentioned in Chapter 1 the V7 tags were chosen for the project making the acoustic signals the type of communication used in the project. This chapter will approach how this communication can be made as well as some sources of error that can affect it.

4.1 Medium and signal modulation

Regardless of which communication technique is used, basic elements are always present. Those are illustrated by Figure 4.1. A message that contains the information to be sent

Table 4.1: Underwater signals summary [40]

	Pros	Cons
Acoustic	Proven technology Range up to 20 km Works in non-line-of-sight	Poor performance in shallow water Affected by turbidity, ambient noise, salinity and pressure gradients
Optical	High bandwidth Low cost	Needs line-of-sight Very short range Susceptible to turbidity
RF	High bandwidths at very close range Works in non-line-of-sight Immune to acoustic noise	Limited range through water Needs high power to have good performance Dependency on water properties (limited results in salt water)

will be created. The message is then codified to travel through the environment and the transmitter will convert the electric signal to an acoustic signal propagating the message through the medium. The medium in this scenario is the underwater environment which is filled with noise, interference and other properties that can compromise the communication. Those factors are going to be mentioned in the next sections of this chapter. Afterwards the acoustic signal is converted to an electric signal by the receiver. This electric signal will be decoded and hopefully the original message will be the same as the one sent by the original device [36], [38].

The acoustic signal can be modulated in various ways depending on the application. However, in this project the signals were already created by VEMCO, which is modulated as a chirp signals. The only information provided by VEMCO was that the signal had a frequency of 69 kHz and an approximate interval between emissions of 90 s and 120 s.

A chirp signal is a signal whose frequency changes over time [42]. These signals are typically used in acoustic modems, since signals that correlate with chirp signals will have a higher value the more similar they are and have small time offsets allowing for more accurate time interval detection. If the frequency rises over time it's called up-chirp and the opposite down-chirp [42], [43]. One example of an up-chirp signal can be seen in Figure 4.2.

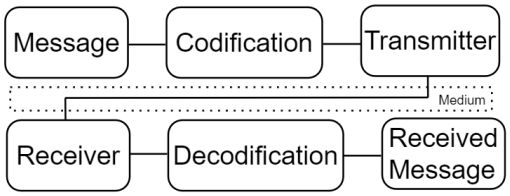


Figure 4.1: Communication diagram

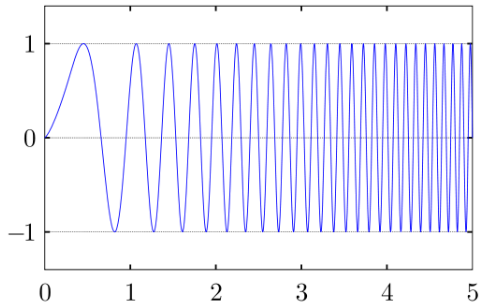


Figure 4.2: Chirp signal example [44]

4.2 Absorption, spreading and noise

As said in [38] the most important property of the underwater acoustic channel is the attenuation. The two main factors that affect this property are the absorption and the spreading loss. An acoustic wave is a mechanical wave and is propagated through the movement of water molecules. The movement of any particle generates heat and the sound travelling through the medium is no exception. The phenomenon of transforming acoustic energy into heat is called absorption [38]. The energy of an acoustic wave is lost over time and the amount of energy that can turn into heat depends in how much the water particles will move/oscillate, in other words the signal's frequency. The frequency of a signal has direct impact in the distance a signal can travel. The distances the acoustic signals can travel and reach the receiver can be divided in very short, short, medium, long and very long. Table 4.2 shows that relation [36]. The relation between the absorption of an acoustic signal and its frequency can be seen in Figure 4.3.

The highest peak of energy of an acoustic signal occurs during the beginning of its transmission. Over time that energy will decrease, as it spreads through the medium, being called spreading loss. This property is the relation between the increase of the surface area and dispersion of the emitted energy. The spreading loss can be spherical or cylindrical. Assuming that the surface area of the emitted signal is spherical and propagates away from a source uniformly in all directions, the rate at which the signal's

Table 4.2: Frequency vs Distance in acoustic signals [38]

	Distance (km)	Frequency (kHz)
Very short	0.1	≥ 100
Short	0.1-1	20-50
Medium	1-10	10
Long	10-100	2-5
Very long	1000	≤ 1

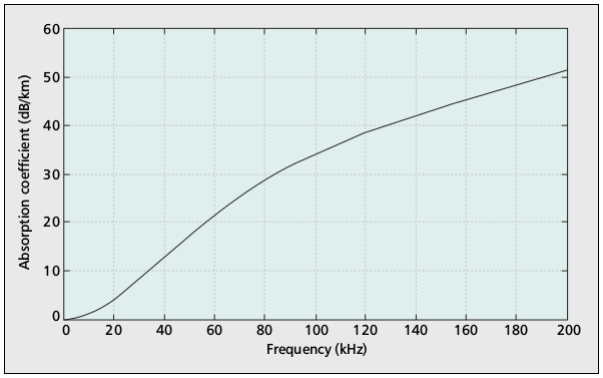


Figure 4.3: Absorption coefficient [38]

intensity decreases can be obtained using the definition of intensity and the principle of conservation of energy [45]. Ideally if the generated acoustic source is radiated equally it must be distributed equally around the sphere. If the distance of the acoustic source increases so does the sphere radius. The amount of power consumed with this can be described with Equation 4.1.

$$P = 4\pi r^2 I \tag{4.1}$$

Where P is the total power, r is the radius of the sphere, and I is the intensity. Intensity is the average amount of sound energy transmitted per unit time through a unit area in a specified direction [45].

Solving for I results in Equation 4.2:

$$I = I_0 \left(\frac{r_0^2}{r^2}\right) \tag{4.2}$$

In [38], [45] says that the intensity decreases as the inverse square of the range for spherical spreading. Assuming that r_0 is 1 m, then at 10 m the intensity of the signal would be 100 times less of the original. An example of spherical spreading can be seen in Figure 4.4.

Another factor that can affect acoustic communications is noise and will always be present in any type of communication. Noise has three main sources: man-made, site-specific and ambient noise [38]. The first one can refer to shipping activities, electronics used near or in the underwater environment, machinery, water sports, military surveillance, among others. Site-specific noises are usually compared to a Gaussian noise, composed of several non-Gaussian signals. Finally the ambient noise is made of turbulence, breaking waves, animal life, marine activity, seismic activity, rain and many others.

Noises that can be described by Gaussian statistics and through a continuous power spectral density are easier to predict [6], [36].

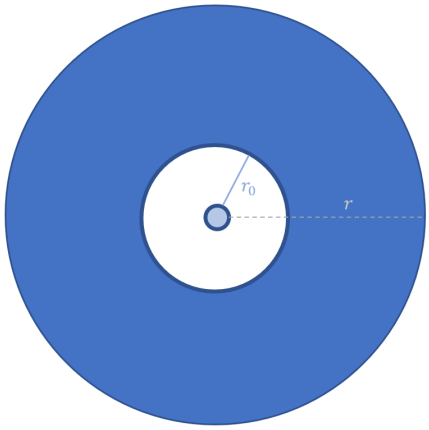


Figure 4.4: Spherical spreading of an acoustic signal

Noise can also be present in the emitted signal, deriving from the electronics used in the emitter and from the frequency of the acoustic signal [38].

4.3 Multi-path

The muti-path is a problem which occurs when the same acoustic signal is received two or more times, since it travelled through more than one path, which generates false-positives as well as receiving the same signal at different times [38], [46]. The multi-path phenomenon can happen when a wave passes through a medium boundary and it can be both reflected and transmitted [46]. A medium boundary is defined as the edge between two different, adjacent mediums.

When a wave bounces on a boundary it gets reflected [46]. Longer distances will be taken when compared to the direct path, called reflected paths. The reflected waves will be attenuated and delayed longer, since they take a longer distance to reach the receiver. Sometimes only reflected waves can reach the target, since the underwater medium can be filled with many obstacles, which can introduce to many errors if a system is trying to compute the position of a target. According to the law of reflection the angle of incidence is equal to the angle of reflection in any reflection [46]. The main causes of underwater acoustics reflections are the water surface, its bottom and bodies in or on it. The number of points of reflection needed to reach a receiver are called order of reflection [46]. An example of this phenomenon can be seen in Figure 4.5, where the direct path is represented as green, with blue a first order reflection and as orange a second order reflection.

Some boundaries/surfaces will be irregular. Due to this, the original wave will get reflected more times when compared with a uniform surface. When this happens many acoustic signals will generate both constructive and destructive interference when received

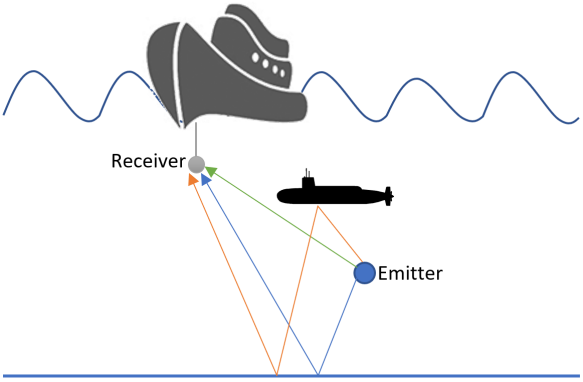


Figure 4.5: Acoustic waves reflection

[38], [46].

When a wave continues to propagate through the new medium it gets transmitted [38], [46]. Most of the underwater environments are not homogeneous, that being said temperature, salinity and pressure will affect how the acoustic waves will propagate and those proprieties will be approached in the Subsection 4.4. Those parameters can change an acoustic wave's trajectory and wiil be refracted. The angle of refraction depends on the sound velocity in two different medium that share a boundary, according to Snell's law. When the trajectory changes the distance to the receiver increases. With this the signal intensity decreases (as Equation 4.2 shows) and it takes more time to reach the receiver. Usually, as shown in Figure 4.7, the parameters that affect acoustic waves' refraction don't vary enough to make a considerable error in low depths[46]. Figure 4.6 has an example of refraction of an acoustic wave where the original trajectory while travelling in medium c_1 changes when it reaches c_2, portraying, for example, when there is a salinity or temperature boundary, which will cause an acoustic refraction.

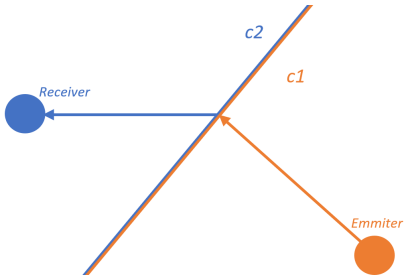


Figure 4.6: Acoustic waves refraction

4.4 Propagation delay and Doppler effect

The delays in underwater acoustic communication systems are appreciable specially when compared with the electromagnetic communications in the air [38]. The sound speed in the water is affected by many parameters such as depth/pressure, salinity and temperature. Estimating the sound speed with minimal errors is necessary to study how this form of communication behaves in the environment [38]. The sound speed velocity in meters per second (m/s), c, can be represented by the MacKenzie Equation 4.3.

$$c = 1448.96 + 4.591T - 5.304 * 10^{-2} T^2 + 2.374 * 10^{-4} T^3 + 1.340(S-35)$$
$$+1.63 * 10^{-2} D + 1.675 * 10^{-7} D^2 - 1.02 * 10^{-2} T(S-35) - 7.139 * 10^{-13} T D^3 \quad (4.3)$$

Where T is the temperature (C°), D is the depth (m) and S the salinity (ppt).

Using the computed velocity in the Equation 4.3, the time a sound needs to travel to a known distance between emitter and receiver can be known with Equation 4.4

$$t = \frac{d}{c} \quad (4.4)$$

Where t is the time of travel (s), d the distance the signal travels and c the sound speed in that medium (m/s).

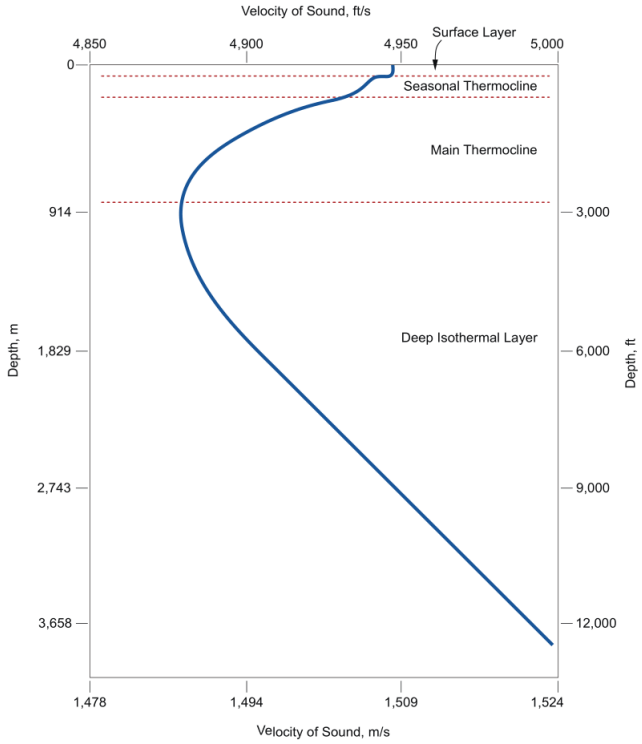


Figure 4.7: Sound Velocity Profile [38]

The Sound Velocity Profile (SVP) makes an analysis in how the sound velocity varies in relation with depth. The SVP also depends on latitude, longitude and the time of the day [38]. Those dependencies increase the uncertainty to the measurements of the speed of sound in these environments.

Observing the Figure 4.7 it is noticeable that near the surface depth almost doesn't affect sound speed in the underwater environment. Afterwards the sound speed decreases in both the seasonal thermodine and main thermodine, which is related with changes in temperature. In the deep isothermal layer, the temperature stabilises and the sound velocity increases. This changes in speed can contribute to the sound wave refraction, as explained in the Section 4.3 [38]. The propagation delay is intimately related with speed of the sound and the medium it travels. If the acoustic wave takes more time than it should to arrive after travelling a certain distance, its velocity or path could have not be the optimum or the estimated. This must be taken into account during signal processing [38].

If there is movement between the emitter and the receiver during the emission of a signal its frequency might shift and/or spread. This phenomenon is called Doppler effect [38]. It is important to take this effect into account since acoustic systems can be affected by their low sound speed (when compared with electromagnetic waves) and drifting. Even the environment can increase the repercussions of this effect, for instance waves and currents can disturb the instruments used in the acoustic system.

This effect can be translated with the Equation 4.5:

$$f_d \;=\; f_0 \,\frac{v}{c} \tag{4.5}$$

Where f_0 is the signal's original frequency (Hz), v is the relative velocity between transmitter (m/s) and receiver and c the sound velocity (m/s).

Chapter 5

Developed Solution

This chapter serves to bridge both the previously contextualised problem and the theoretical study with the implemented solution. Many positioning systems were studied in Chapter 3. Most of those solutions depend on both target and tracker to be able to receive and send acoustic signals or using many receivers. The developed acoustic tracking solution relied on an algorithm inspired in DoA solutions, which is described in detail in Chapter 7, since the main technical goal is to detect the direction of the acoustic signal relative to the boat, without knowing the range to the target.

Figure 5.1 has a representation of the developed solution's high-level architecture. First of all the target's acoustic signal needs to be detected and identified. That is done with algorithm that was present in [3] and explained in Chapter 6. The proposed solution, uses three channels for data acquisition, composed with the hardware mentioned in Chapter 9. Given the dynamics of the V7 tags, that is, their rather long emission rate, it has become difficult not only to test the system, but also to estimate a position with so few measurements. That said, an emission system has been developed which, besides having a larger signal amplitude, its emission rate is controllable. The hydrophones will be displayed in a baseline that forms an equilateral triangle, which guarantees equal angular distances. This configuration choice is explained in the Chapter 7. Each of those channels is connected to a board which is responsible to synchronise them, at every second. Everything in Figure 5.1 diagram, after the channels, is processed by the computer. This allows the microcontroller (uC) to only spend time in data acquisition and for the preparation of packages to be sent via Ethernet, avoiding data loss.

After detecting and having the ID of any tag that might have emitted its signal in the acoustic channel, if the detection is successful in the three channels, the synchronisation process will be made, which is also explained in Chapter 7, minimising errors concerning the times of arrival of the acoustic signal in each channel. With the detections synchronised, the time of arrival of each channel has the same time reference and the DoA of the acoustic source can be computed with Algorithm 3. The DoA algorithm has its setup

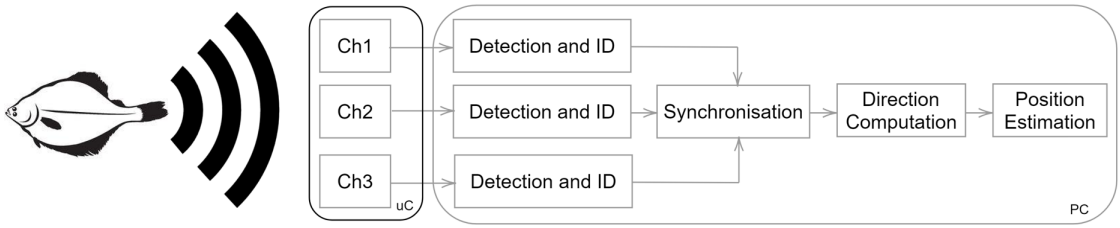


Figure 5.1: High-level architecture

of receivers linked to the same computer using a synchronised time source. This enables the computation of the DoA to be made in the ASVs frame of reference. The different arrival times that each receiver identifies a signal are compared with one another. With that data the angle in relation with the three receivers is computed. Depending on the receivers configuration, the attained angle can be the bearing the ASV needs to have to follow a target. To validate if the developed DoA algorithm could be applied in a tracking application, an error analysis was made, addressed in Chapter 7.

While a solution was being developed, two application scenarios emerged which used the DoA outputs to estimate a target's position besides tracking it. Those scenarios are described in more detail in Chapter 8. The first one relies on a moving ASV/boat with a setup of three receivers. After some computed directions it starts to orbit the selected target. The second scenario uses a gate on the river, with two setups with three receivers each in a buoy. In this scenario an assumption is made that each buoy can communicate their computed data with one another. Both methods were studied and validated with a simulator developed from scratch, with an incorporated Kalman Filter.

Figure 5.2 summarises the problem. An emitter with an unknown position with coordinates x_t and y_t sends a codified signal. That signal is then detected and identified by each of the three receivers in the ASV composing the baseline b, with known coordinates given by GPS, x_b, y_b and z_b. Each channel, ch_1, ch_2 and ch_3, identifies the signal at a given time t_{ch_1}, t_{ch_2} and t_{ch_3}, respectively. With those time values a DoA is computed. Depending on the scenario, the position will be estimated with different methods. The boat scenario will describe an orbit around the target, which will enable the interception of the lines attained with the DoA of consecutive signal identifications. That interception estimates the coordinates of the point where the target might be. The error will be bigger the higher the target's speed. The buoys scenario assumes that they communicate in real-time with each other. Each buoy will be able to compute a DoA related with themselves and the target. As with the boat scenario, this allows the computation of the equation of a line with the attained angle. In this case, however, since two different lines are attained for the same signal identification, the position can be estimated without relying on the next

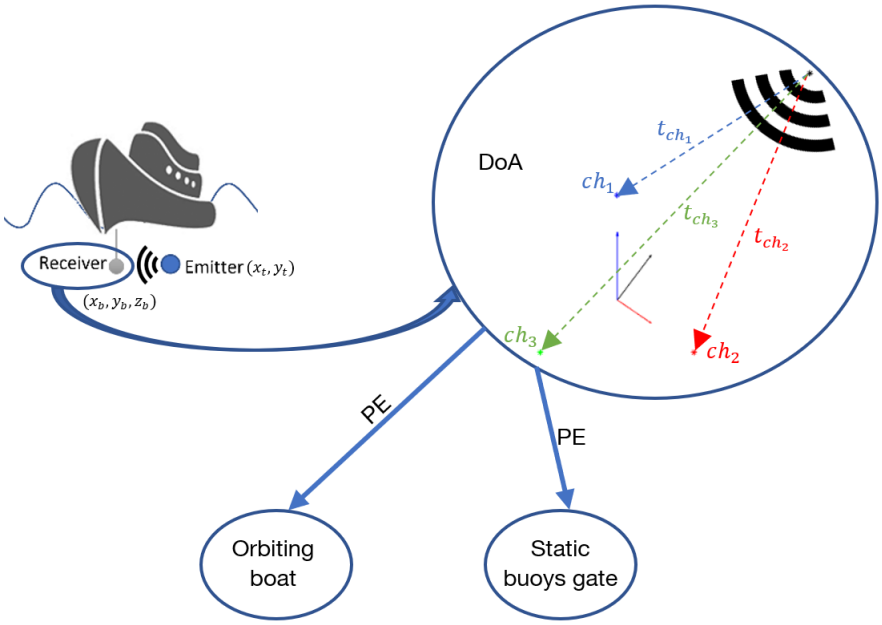


Figure 5.2: DoA and tracking/PE system

detection. The target speed won't affect this method, but since the buoys are stationary, the target can't be followed. To solve this, more pairs need to be deployed in the mission area.

THIS PAGE
INTENTIONALLY
LEFT BLANK

Chapter 6

Acoustic signal detection & identification

This chapter addresses the method that was previously presented in [3] concerning the detection and identification of acoustic signals.

In the first phase of analysing the tags codification and the initial development the hardware that was used is the one that was mentioned in Chapter 2. For recording acoustic signals the AS-1 hydrophones were used, connected to the PA-4 preamplifiers, both from Aquarian Audio. Then they were connected to the soundboard ZOOM UAC-2. The acoustic signals in [3] were recorded at a rate of 192 kHz. The signals emitted by a V7 tag were recorded. These signals are composed of pings such as the one in Figure 6.1 which resembles signals modulated with chirp modulation, mentioned in Chapter 4 . More than one tag was recorded to understand more about their signals. All the recorded tags have 8 pings and for each tag the pings are sent always spaced with specific time intervals. One example of a tag signal can be observed in Figure 6.2. Each signal was sent with an interval ranging from 90 s to 120 s. All the recorded tags had different time intervals between each ping. So it came to conclusion that the time intervals between pings was how the tags could be identified, in other words their ID. However the first interval was common in all tags, so to distinguish tags the next 6 intervals are necessary.

Table 6.1: Intervals in samples between pings of the V7 tags at a sampling rate of 500 kHz

Interval / ID	$\Delta 1$	$\Delta 2$	$\Delta 3$	$\Delta 4$	$\Delta 5$	$\Delta 6$	$\Delta 7$
1308	140112	230013	210005	260021	169990	230013	190063
1309	140047	269992	200034	170026	189997	150018	159953
1310	140083	239992	290008	160063	190005	149990	270000

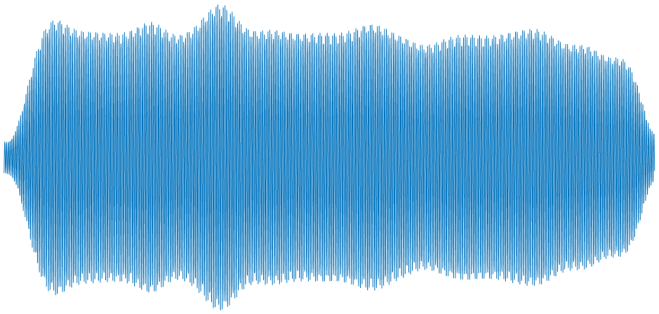


Figure 6.1: Ping signal

The tags were logged by simply recording them many times and by introducing the average of each interval in a database. Examples of logged tags can be seen in the Table 6.1. The table contains the ID and the respective intervals in samples, with an acquisition rate of 500 kHz. No other information was possible to attain from the tag's codification. The detection and identification algorithm is based on the known time intervals. When a signal is recorded if a tag signal was received it would be compared with the database in order to identify which was the tag. For that to be possible, the pings needed to be found in an audio recording. Algorithm 1 is responsible of doing so. The signal is filtered via software with a Butterworth passband filter, with a low cutoff frequency at 71 kHz and the high cutoff frequency at 67 kHz, to mainly accept the desired frequencies of the tags signals.

The filtered recorded signal is then correlated with the samples of a recorded ping (the signal to be found in the recordings), which is going to be called identifier signal. If any ping exists in the recording high peaks will result from the correlation. Where the correlation has the highest values is where the biggest match with the identifier signal is found, being the most probable instant when a ping was received. The correlation results are seen as C in Algorithm 1. The root mean square (RMS) of C is computed. A threshold is imposed by the RMS and its multiplier, R_M and the latter can be chosen by the user.

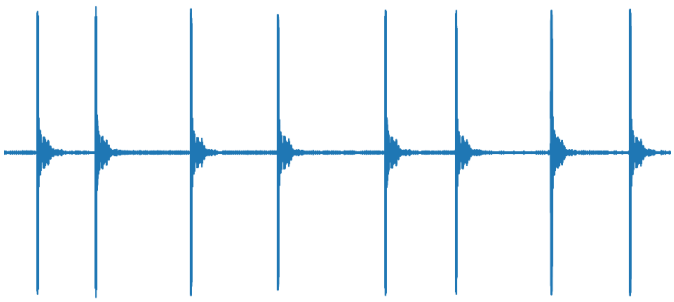


Figure 6.2: V7 tag signal example

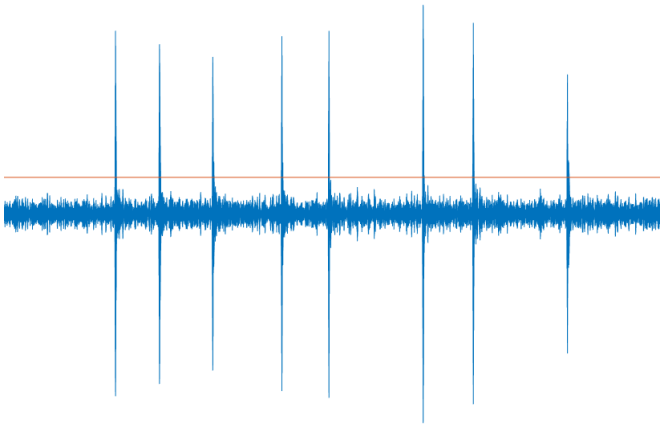


Figure 6.3: Threshold applied to the Correlated recorded signal

The array Q is filled with indexes of the samples that are above that threshold. This process is illustrated by Figure 6.3.

Two arrays are created, saving the beginning and the end of each ping, P_b and P_e, respectively, in lines 7 to 17 of Algorithm 1. The size of either P_b or P_e arrays corresponds to the number of possible pings that are in that recording. Finding the maximum value for each possible ping is done in lines 18 to 19. Within the gap that is composed of P_b and P_e, for the same index, there are values that are possible locations for that ping. Finding the maximum value in that gap indicates the index where the ping is at, P.

Afterwards, the intervals between possible pings are computed, ΔP. Computing and storing those intervals is useful, since each tag has a specific ID that is identified with intervals between its pings, as referred in this section. However, identifying tags isn't possible with only those intervals since tags can overlap each other. Also, some samples will pass through the threshold without any pings, creating the need of changing variables, such as the threshold and margin, in order to avoid the risk of losing data. To detect and identify tags taking those issues into consideration, Algorithm 2 was developed.

Lines 1 to 18 of Algorithm 2 are responsible for finding the beginning of a tag, checking each value of ΔP with the use of the variable D. The margin, M, is used to ensure an interval of acceptance for the difference between the first two pings of a tag, called starting interval, S_Δ. If any value of D is above or below the S_Δ, taking into account M, it won't be accepted as a start of a tag. If it is accepted as a beginning of a tag, a set of control variables are saved, lines 11 to 15, in matrix, $Matrix$.

Next, to each logged tag, the remaining logged intervals will be checked, taking into account the defined margin of acceptance M, alongside the intervals that were found in the recordings. If 6 of them were found, it means the 7 intervals were identified and, therefore, a tag was also found, saving the time-stamps of each ping and its identification in the matrix T_{ID}. By checking the intervals not only to the next ping but also to the

following pings, it is possible to identify a tag even if it is among noise that got through the threshold, or if tags overlapped each other in that recording. Having the 8 different detected ping time stamps saved at each tag detection, instead of just one indicating the tag's general time stamp, provides a more efficient way of a possible bearing estimation, since that will originate 8 different time of arrival differences for each tag.

Algorithm 1 Find pings Algorithm

Input: Cross-correlation samples, C, gap size, G, and RMS multiplier, R_M.

Output: Pings, P, and pings intervals, ΔP.

Compute root mean square, RMS, of C.
for $i = 0$ to size of array C **do**
  if $(C[i] > RMS * R_M)$ **then**
    fill array Q with i.
  end if
end for
Set ping begin, P_b, and ping end, P_e arrays.
$n_b \leftarrow 1, n_e \leftarrow 0$.
$P_b \leftarrow Q[0]$.
for $i = 0$ to (size of array Q -1) **do**
  if $(Q[i+1] - Q[i] > G)$ **then**
    $P_b[n_b] \leftarrow Q[i+1]$.
    $P_e[n_e] \leftarrow Q[i]$.
    $n_b \leftarrow n_b + 1, n_e \leftarrow n_e + 1$.
  end if
end for
$P_e[n_e] \leftarrow Q[i+1]$.
Find the indexes where the correlation values are higher between each P_b and P_e.
Array P is filled with the index of the maximum value found at each interval.
Compute the intervals between consecutive pings, ΔP.
return P and ΔP.

Algorithm 2 ID tags

Input: Pings, P, delta pings, ΔP, tags logs, T, and margin, M.

Output: tags' ID, T_{ID}.

for $i = 0$ to size of array ΔP **do**
  for $h = 0$ to size of array $(\Delta P - i)$ **do**
    if (h == 0) **then**
      $D \leftarrow \Delta P[i]$.
    else
      $D \leftarrow \Delta P[i + h] + D$.
    end if
    if $D > S_{\Delta} + M$ **then**
      break.
    end if
    if $(D < S_{\Delta} + M$ and $D > S_{\Delta} - M)$ **then**
      $Matrix[1][j] \leftarrow i, Matrix[2][j] \leftarrow D$.
      $Matrix[3][j] \leftarrow h$.
      $j \leftarrow j + 1$.
      break.
    end if
  end for
end for
for $i = 0$ to size of $Matrix[2]$ **do**
  $found \leftarrow 1, k \leftarrow (Matrix[1][i] + Matrix[3][i])$.
  for y = 0 to the number of logged tags **do**
    for j = 1 to 6 **do**
      for $h = 0$ to size of $(\Delta P - k)$ **do**
        if (h == 0) **then**
          $D \leftarrow \Delta P[k + 1]$.
        else
          $D \leftarrow \Delta P[k + h + 1] + D$.
        end if
        if $D > (T[j][y] + M)$ **then**
          $found \leftarrow 0$.
          break.
        end if

```
            if (D < T[j][y] + M and (D > T[j][y] − M) then
              M_D[j − 1] ← D, k ← k + h, found ← 1.
              if (j==6) then
                The identified TAG is stored as long as its pings' time stamps, T_ID.
              end if
            else
              found ← 0.
            end if
          end for
          if (found == 0) then
            k ← Matrix[1][i] + Matrix[3][i], break.
          end if
        end for
        if (found == 1 and j == 6) then
          break.
        end if
      end for
    end for
    return T_ID.
```

THIS PAGE
INTENTIONALLY
LEFT BLANK

Chapter 7

Direction of Arrival Algorithm

This chapter addresses the developed algorithm for the DoA of the tag's acoustic signal. Not only that but also an error analysis for the algorithm's returning results, as well as the synchronisation algorithm for error minimisation when concerning the arrival times in each channel.

7.1 DoA algorithm

As said in Chapter 3 the target won't do more than pinging a signal. Most of the solutions depend on knowing the target's depth, the signals' time of travel and the use of a big array of hydrophones, so an alternative solution was found which would allow to track targets under these conditions at a low cost and for that a bearing or DoA algorithm was developed. The solution will be able to track a target without knowing the initial range to it and, eventually, estimate its position.

For the algorithm to work properly the hydrophones placement is important, since it is one of the main factors for the relation of the arrival times between them. Three hydrophones are used and are placed at equal angular distances (i.e. 120 °) and equal linear distances from each other, thus resulting in an equilateral triangular baseline. Each hydrophone is assigned a vector placed at the baseline's centre and pointing in that hydrophone's direction, and the magnitudes of each vector are determined by the arrival time. The magnitudes are normalised such that the hydrophone that first captures the incoming acoustic signal has a magnitude of 1, whereas the last hydrophone to do so has its vector's magnitude assigned to a value of 0. Using the vectorial sum of the three vectors, the approximate incoming angle of the acoustic source can be determined, as shown in Figure 7.1 a). Because of the symmetrical shape of the baseline, whenever the source is perfectly aligned with any given hydrophone and the baseline's centre, two of the vectors cancel out, thus resulting in a perfect estimation of the incoming angle, as demonstrated in Figure 7.1 b).

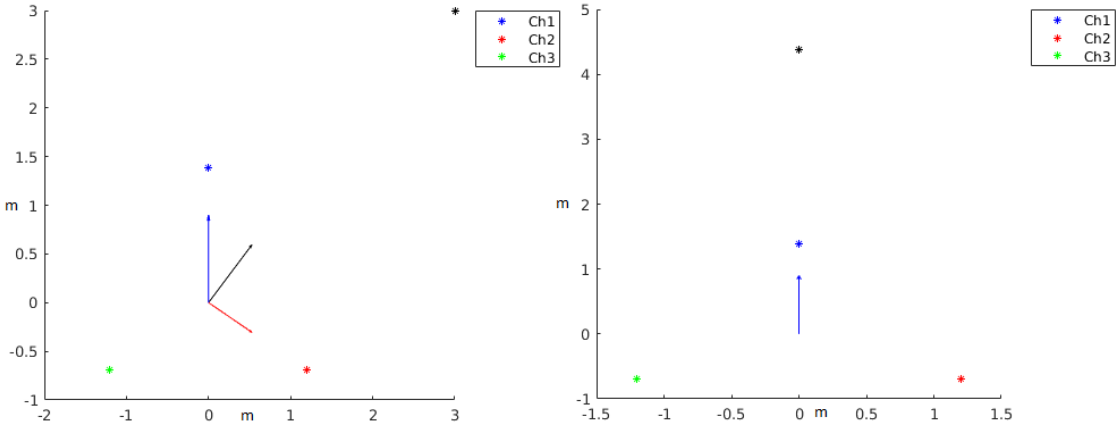


Figure 7.1: a) Vectorial sum of the three vectors and b) Acoustic source aligned with one channel

Guaranteeing they are placed as said above, the algorithm expects to receive the tag's synchronised IDs. which guarantees the identified samples have minimal error, that is done by Algorithm 5. The used sample for the arrival time is the first detected ping of the found tag.

Algorithm 3 summarises the solution. Lines 1 to 7 are responsible of creating a baseline with equal angular distances, referencing the channels to the baseline centre, b_c, depending on the equilateral triangle side, l.

The arrival time of each channel will be stored at a_{time}. Each sample will be subtracted with the minimum (fastest) amongst them, seen as the time difference of arrival between channels, *tdoa*. This process is done through lines 8 to 21.

The values are normalised, guaranteeing that the highest value will correspond to the channel with the fastest detection. This is done from line 22 to 26. With the normalised values the vectors are generated and each vector has two components: magnitude and direction. The vectorial sum of the three vectors will result in an array, $result_{array}$, which has a magnitude and direction, Θ. This direction is the desired target's DoA.

Algorithm 3 Direction of acoustic source algorithm

Input: Synchronised IDs, $channel\_1$, $channel\_2$, $channel\_3$.

Output: Angle theta, Θ, with tag ID, ID

1: Original angles, $o\_angles \leftarrow [90, 240 + 90, 120 + 90]$ and baseline, $baseline \leftarrow [ch1, ch2, ch3]$
2: calculate the baseline's centre, b_c, and how the channels $ch1$,$ch2$,$ch3$ are disposed, where l is the side of the equilateral triangle
3: $b_c \leftarrow \frac{\sqrt{3}}{3} * l$
4: **for** $i = 0$ to 3 **do**
5: $\quad [x, y] \leftarrow pol2cart(deg2rad(o\_angles[i]), b_c)$
6: $\quad baseline[i] \leftarrow [x, y, z]$
7: **end for**
8: $a_{time} \leftarrow [channel\_1[1], channel\_2[1], channel\_3[1]$ saving the sample where the detection was done in each channel
9: subtract each sample where the detection was made in each channel, with the minimum (fastest) amongst them
10: **for** $i = 0$ to 3 **do**
11: $\quad tdoa(i) \leftarrow a_{time}(i) - min(a_{time})$
12: **end for**
13: the $interval$ array stores the $tdoa$ values in descend order, and the $angle\_index$ contains their original positions
14: $[interval, angle\_index] \leftarrow sort(tdoa)$
15: **for** $i = 0$ to 2 **do**
16: $\quad interval(i, 0) \leftarrow -interval(i)$
17: **end for**
18: **for** $i = 0$ to 3 **do**
19: $\quad interval(i, 1) \leftarrow o\_angles(angle\_index(i))$
20: $\quad interval(i, 2) \leftarrow angle\_index(i)$
21: **end for**
22: $b \leftarrow 1$ and $m \leftarrow -b/interval(0, 0)$
23: normalise the values with the previous equation, where the highest value will correspond to the channel where the acoustic signal was the first to arrive
24: **for** $i = 0$ to 3 **do**
25: $\quad interval[0][i] \leftarrow m * interval[0][i] + b$
26: **end for**
27: generate the vectors from the normalised values, with their direction and magnitude
28: $[x, y] \leftarrow pol2cart(deg2rad(interval(:, 2)), interval(:, 1))$

29: vectorial sum of the three vectors
30: $result_{array} \leftarrow [sum(x), sum(y)]$
31: transform the Cartesian (x,y) coordinates to obtain the vector's direction
32: $[\Theta, rho] \leftarrow cart2pol(result_{array}(1), result_{array}(2))$;
returnΘ, ID.

7.2 Error analysis

To validate the algorithm exhaustive tests were made. A simulation for the error analysis was developed in Matlab.

The baseline would be in a fixed position, varying where the target would emit its signal. The target would vary its angle, its depth and its distance in relation to x and y, which was called magnitude in m. All of those parameters were varying in relation to the baseline's centre.

The depth changes with a step of -5 m until reaching -20 m . For each value of depth, the magnitude varies from 3 m to 300 m, with a step of 1 m. For each of the magnitude values there is also an angle which varies from 0 to 360 ° with a step of 1 °.

The Algorithm 4 is an adaptation of Algorithm 3 which uses the known target coordinates, given by x_{target}, y_{target} and z_{target}, to simulate the a_{time}. This can be given with Equation 7.1, where T_k are the target's coordinates, ch_k the channel coordinates and v_{water} the sound velocity in the water. This equation is applied to each channel. However

Algorithm 4 DoA error analysis

Input: Synchronised IDs, $channel\_1$, $channel\_2$, $channel\_3$.

Output: The maximum angle error, $maximum_e$, and the mean of the results' errors, $mean_e$

1: **for** $z_{target} = 0$ to -20 **do**
2: **for** $magnitude = 3$ to 300 **do**
3: **for** $angle = 0$ to 360 **do**
4: $[x_{target}, y_{target}] \leftarrow pol2cart(deg2rad(angle), magnitude)$
5: $[\Theta, \Theta_e] \leftarrow DoA(x_{target}, y_{target}, z_{target}, a_{time_e})$
6: Save the results, $results$, from the function DoA at each iteration
7: **end for**
8: **end for**
9: **end for**
10: Compute the mean and the maximum error out of all iterations, $mean_e$ and $maximum_e$, respectively
return$mean_e$,$maximum_e$.

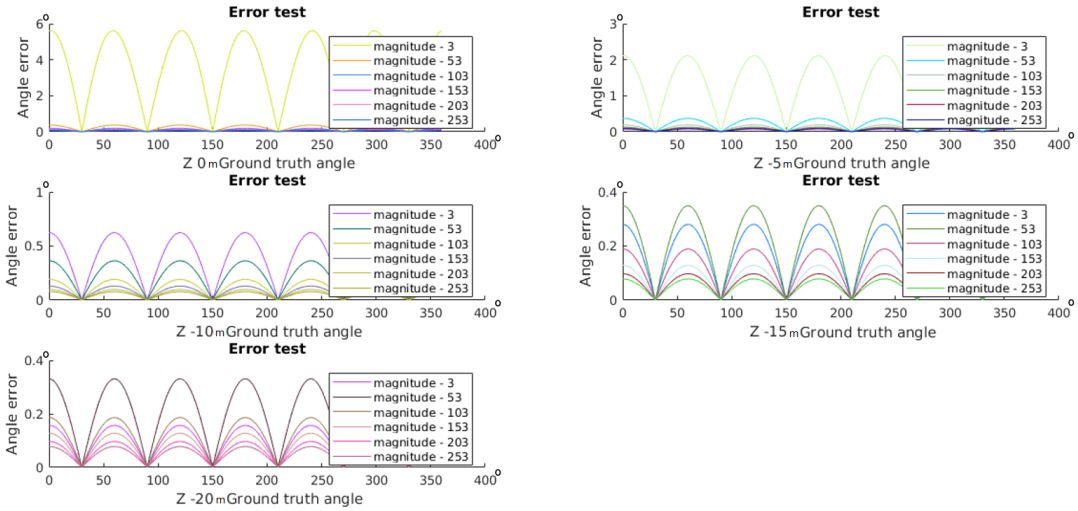


Figure 7.2: DoA error analysis with no arrival time error

this equation already takes into account that the a_{time} in each receiver can have delays. This simulation adds a random arrival time error, $a_{time_{error}}$, to the nominal value, which ranges between $1 * 10^{-}7$ ns to 50 us.

$$a_{time} = \sqrt{\frac{(T_x - ch_{k_x})^2 + (T_y - ch_{k_y})^2 + (T_z - ch_{k_z})^2}{v_{water}}} + a_{time_{error}} \tag{7.1}$$

This allows to study all the parameters that can affect the results of the DoA algorithm, including the arrival time at each channel. Knowing this, 537890 different scenarios were studied for each of tests, with and without the $a_{time_{error}}$. Since both the target and the baseline positions are known by the simulator, the nominal values can be computed and then be used to be compared with the values attained through the DoA algorithm. The results without $a_{time_{error}}$ can be seen in Figure 7.2.

All the depth scenarios are shown as different subplots. The target angle, in °, varies in the x axis for the correspondent magnitude, in m, which are represented in different colours, as the plot legend shows. The error is represented in the y axis. It is noticeable that the error is bigger the closer the target is to the baseline, decreasing the error as the magnitude and depth increases.

The results with the $a_{time_{error}}$ are represented in Figure 7.3. The same can be said about the results attained from this test, the closer the baseline is to the target, the higher the errors. The plots can show noise throughout their representation, since time delays are added to simulate what could happen in real-life. With proper hardware and software tests, the error found in real-life scenarios can be inserted in this simulation, highlighting the delays of arrival times between channels, detectability range limits and many others. Table 7.1 contains the mean and maximum errors for both tests with all iterations taken

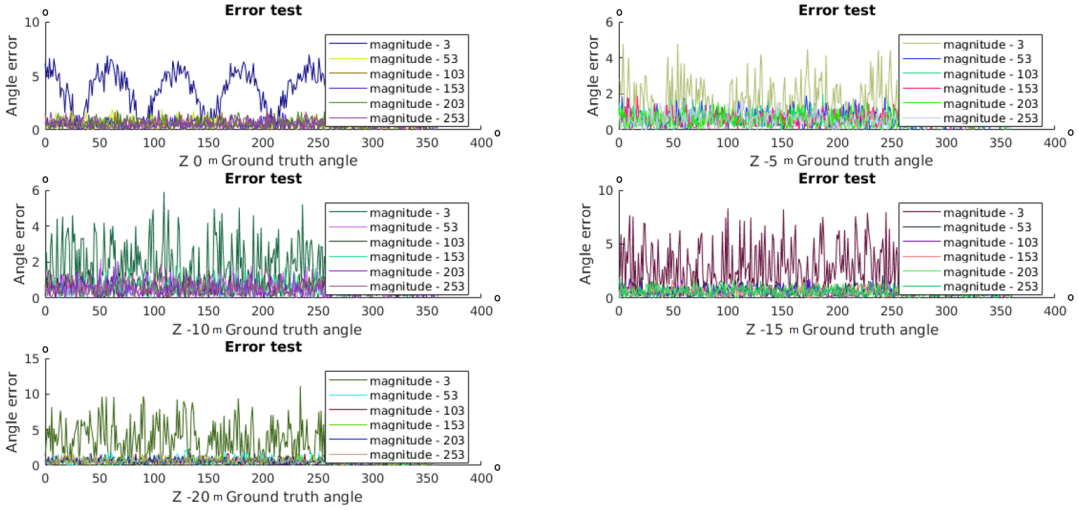


Figure 7.3: DoA error analysis with arrival time error

into consideration. The results are positive, even if the maximum error with the added delays is considerable, however observing the subplots of Figure 7.3 and the mean value, values like that are an exception when compared with the remaining iterations.

Table 7.1: DoA error analysis

test	mean	maximum
no delay	0.1528 °	5.6108 °
delay	0.6408 °	11.3938 °

7.3 Synchronisation algorithm

The DoA algorithm needs to have the channels synchronised to minimise the errors related with the arrival times in each channel. For this purpose a trigger system was used which will be explained in more detail in Chapter 9. Algorithm 5 addresses this issue. Any of the $channel_x$ arrays has a tag ID, its pings and the last two positions 9 and 10 are the nearest triggers in relation with the first ping of the detected tag, being the ninth the closest to it. The algorithm assures that all channels are referenced to one of them, in this case channel 1. The differences between the first ping of the detected signal and the last two nearest triggers related to it are computed being represented, for example, as $t11$ in Figure 7.4. The algorithm's threshold is imposed by the maximum interval that is possible to have between channels in the baseline. Afterwards the intervals between the channels and the nearest trigger are computed, lines 4-6. If all the intervals are bellow the

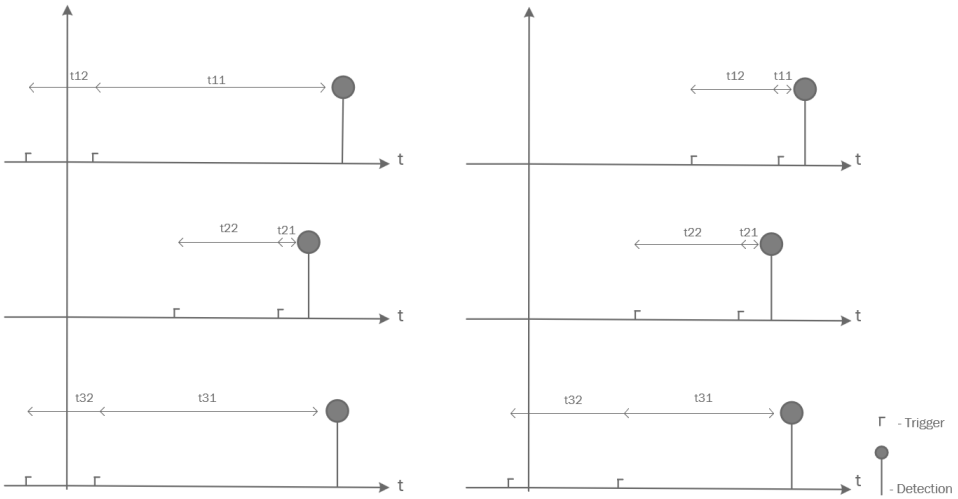


Figure 7.4: Synchronisation situations a) and b)

threshold every channel is referenced to the nearest trigger in relation to the first ping, lines 7-14. If that's the case, all of the values should be referenced to intervals computed in lines 15 - 17. Since channel 1 was chosen to reference the data, only channel 2 and 3 values need to adjust their time frames, adding the intervals Δ_{12} and Δ_{13} to their values, respectively.

Sometimes the channels are not referenced to the same trigger, as can be seen in Figure 7.4. One of the situations is exposed in Algorithm 5 and observable in Figure 7.4 situation a). The differences between the first ping of detection and the nearest trigger aren't the same in all channels, being smaller in the second one. The intervals where the second channel is common, Δ_{12} and Δ_{23}, will have their values above the threshold. So when a channel fails two of the intervals where it belongs will automatically fail, being easy to know what's the issue in the synchronisation, as seen in line 25. To deal with this, instead of using the first nearest trigger in the channel that failed, the second nearest trigger will be used. The threshold test is repeated with the new values. If it still fails then the other two channels will also change to their second nearest trigger, guaranteeing that all the channels are synchronised. After that is done, the triggers are removed from each channel array, lines 42-44.

Algorithm 5 Synchronisation Algorithm

Input: Time difference between the 1st ping and the two previous triggers $t11$, $t12$, $t21$, $t22$, $t31$, $t32$, and each channel detected signal *channel_1*, *channel_2*, *channel_3*.

Output: Synchronised detections, *channel_1*, *channel_2*, *channel_3*.

compute threshold, *threshold*, where v_s is the speed of sound in the water and b_s,the baseline side
$threshold \leftarrow b_s/v_s$
compute differences between the channels and nearest trigger
$\Delta_{12} \leftarrow |t11 - t21|$
$\Delta_{13} \leftarrow |t11 - t31|$
$\Delta_{23} \leftarrow |t21 - t31|$
if $(\Delta_{12} < threshold)$ **then**
    $okay_{12} \leftarrow 1$
else
    $okay_{12} \leftarrow 0$
end if
the same is done for $okay_{13}$ and $okay_{23}$
if every channel is referencing to the nearest trigger
if $okay_{12} == 1$ and $okay_{13} == 1$ and $okay_{23} == 1$ **then**
    $\Delta_{12} \leftarrow channel\_1[9] - channel\_2[9]$
    $\Delta_{13} \leftarrow channel\_1[9] - channel\_3[9]$
    $\Delta_{23} \leftarrow channel\_2[9] - channel\_3[9]$
    reference to channel 1
    for $i = 1$ to size of array *channel_1* **do**
        $channel\_2[i] \leftarrow channel\_2[i] + \Delta_{12}$
        $channel\_3[i] \leftarrow channel\_3[i] + \Delta_{13}$
    end for
end if
if the sync trigger isn't the same in channel 2, as shown in case a) in Figure 7.4
if $okay_{12} == 0$ and $okay_{13} == 1$ and $okay_{23} == 0$ **then**
    $\Delta_{12} \leftarrow |t11 - t22|$
    $\Delta_{23} \leftarrow |t22 - t31|$
    test if they've been accepted by *threshold*

29: **if** $okay_{12} == 1$ and $okay_{13} == 1$ and $okay_{23} == 1$ **then**
30: $\Delta_{12} \leftarrow channel\_1[9] - channel\_2[10]$
31: $\Delta_{13} \leftarrow channel\_1[9] - channel\_3[9]$
32: $\Delta_{23} \leftarrow channel\_2[10] - channel\_3[9]$
33: use lines 19-22 again
34: **else**
35: $\Delta_{12} \leftarrow channel\_1[10] - channel\_2[9]$
36: $\Delta_{13} \leftarrow channel\_1[10] - channel\_3[10]$
37: $\Delta_{23} \leftarrow channel\_2[9] - channel\_3[10]$
38: use lines 19-22 again
39: **end if**
40: **end if**
41: the same would be done for each channel that would have a different reference trigger
42: $channel\_1 \leftarrow channel\_1[0:9]$
43: $channel\_2 \leftarrow channel\_2[0:9]$
44: $channel\_3 \leftarrow channel\_3[0:9]$
return $channel\_1$, $channel\_2$ and $channel\_3$.

THIS PAGE
INTENTIONALLY
LEFT BLANK

Chapter 8

Position estimation

In order to validate whether the developed DoA algorithm could be implemented in a real situation, i.e in an ASV, it had to undergo further testing than those noted in the previous Chapter 7. For this a simulator was developed composed by two different methods. This chapter presents an exposition of the capabilities that the simulator has, from the parameters it controls as well as the scenarios it can simulate. In addition, topics such as the *Kalman* Filter, whose implementation would ensure better results, are also covered.

8.1 *Kalman* Filter

The arrival time in each channel, as was tested in Chapter 7, is subjected to delays. In addition under certain conditions and with the proposed methods it is not always possible to estimate a position. Knowing this a linear *Kalman* Filter (KF) was implemented in order to minimise those issues.

8.1.1 Linear *Kalman* filter theory

The linear KF is a recursive algorithm which processes data and allows the estimation of states in a system that has noise [47], such as the one in this project. The filter assumes that all state variables that make it up are Gaussian and random. Not only that but also that the states and measurements are described by linear systems, thus being described described by linear equations. Each variable has a weighted average u, which is the centre of the Gaussian distribution, and a covariance, σ^2 which represents the uncertainty [47].

The linear KF can be splited into two main phases: the predict and update functions. The predict phase estimates the state variables from the past iteration and the update phase allows the measurements made from sensors (observations) to update the predict's estimation. For the filter to work properly it needs to be initialised. That can be done by either making an initial guess for the initial state, x_0, or by using the first measurement.

The initial covariance matrix, P_0, also needs an initial estimate. The time the KF will take to converge will depend of those values [47].

After the filter is initialised, until no measurements are made, the predict phase will estimate the current state $\hat{x}_{k|k-1}$ depending on the last iteration. This action can be described by Equation 8.1:

$$\hat{x}_{k|k-1} = A_k \cdot \hat{x}_{k-1|k-1} + B_k \cdot u_k \tag{8.1}$$

Where A_k is a matrix which describes the system's dynamic model and propagates the previous state to the current state, B_k and u_k the matrix and control vector, respectively, which describe external inputs in the system. In this phase the current covariance matrix, $P_{k|k-1}$, is estimated:

$$P_{k|k-1} = A_k \cdot P_{k-1|k-1} \cdot A_k^T + Q_k \tag{8.2}$$

Where Q_k is the noise associated with the propagation model and the rest of the Equation 8.2 refers to the uncertainty propagation of the previous state to the current state.

The update phase starts when new measurements occur in the system. If that's the case the estimated state and covariance from the predict phase will be used to compute the new state vector alongside the covariance matrix.

To make a relation between the last estimated states and the measurement/observations in order to test the KF consistency, the innovation, $\tilde{y}_k$, is computed:

$$\tilde{y}_k = z_k - H_k \cdot \hat{x}_{k|k-1} \tag{8.3}$$

Where matrix H_k represents the system's observation model and it is responsible for making a relation between the observations and the previous state, since not all variables can be referenced to the same units, allowing the conversion from state space to observation space. z_k is the current observation and $\hat{x}_{k|k-1}$ the last known state, estimated in 8.1. After some iterations the innovation value should have an average near to 0.

In order to weigh the estimation of the predict phase with the current observations, it is necessary to calculate the *Kalman* gain, K_k. This gain allows to define the confidence that is given to each of the phases, i.e the lower the resulting value the better the predicted values and vice versa. To compute the *Kalman* gain, Equation 8.4 is used:

$$K_k = P_{k|k-1} \cdot H_k^T \cdot S_k^{-1} \tag{8.4}$$

Where $P_{k|k-1}$ is the last estimated covariance in the predict phase, computed in Equation 8.2 and S_k is the innovation's covariance, given by:

$$S_k = H_k \cdot P_{k|k-1} \cdot H_k^T \cdot R_k \tag{8.5}$$

R_k is the noise associated with the observation model, generally related with the sensor or method that outputs the measurement. Having all those values computed, the new state, $x_{k|k}$, can be computed in the update phase:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \cdot \tilde{y}_k \tag{8.6}$$

As well as the covariance matrix:

$$P_{k|k} = (I - K_k \cdot H_k) \cdot P_{k|k-1} \tag{8.7}$$

Where I is the Identity matrix which has the same size as the number of states in the system.

8.1.2 Linear *Kalman* filter implementation

The targets in the MYTAG project or other emitter in the underwater environment will mostly have slow dynamics, resulting in low velocities which can help in the assumption that the system is linear. Being linear the target movement can be described by:

$$\begin{cases} x_k = x_{k-1} + v_x \cdot dt \\ y_k = y_{k-1} + v_y \cdot dt \end{cases} \tag{8.8}$$

Where $[x_k, y_k]$ are the target's position in m in the instant k, $[x_{k-1}, y_{k-1}]$ in the past iteration, $[v_x, v_y]$ the target's velocity in instant k in m/s and d_t the time interval between k and $k-1$ in s. The z coordinate isn't considered since it isn't relevant for the tracking solution. The state space vector, $\hat{x}$ is given by:

$$\hat{x} = [x \ y \ v_x \ v_y] \tag{8.9}$$

The filter initialisation can be given by the first measurement or any initial user defined values .

As seen in Equation 8.1 the predict phase needs matrix A_k so the previous state can be propagated to the current state. For this system the dynamics matrix is given by:

$$A_k = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.10}$$

In Equation 8.1 there is also the B matrix that introduces external inputs to the system, such as wind, waves, and many others. However, it wasn't taken into account. Q_k matrix is needed for the predict's covariance computation, Equation 8.2. Each value in the matrix's diagonal is a correspondent uncertainty to each of the state's. The choice of

these values is crucial for the filter's convergence and the most difficult to be calibrated. This matrix is given by:

$$Q_k = \begin{bmatrix} \sigma_x & 0 & 0 & 0 \\ 0 & \sigma_y & 0 & 0 \\ 0 & 0 & \sigma_{v_x} & 0 \\ 0 & 0 & 0 & \sigma_{v_y} \end{bmatrix} \tag{8.11}$$

The measurements are given by a method and not by a sensor directly, since only acoustic signals are measured directly. The measurements of the system are the target's coordinates in x and y in m. Only two states are directly observable in the system, since the target's velocity isn't measured. The sensors measurements are given by z_k:

$$z_k = \begin{bmatrix} z_x \\ z_y \end{bmatrix} \tag{8.12}$$

To convert the states vector to the observation model the H_k matrix is used:

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{8.13}$$

For last, so that the update phase can be done the R_k matrix is needed. This matrix represents the observation model and contains the noise that is associated to the sensors and/or used method. The values in this matrix's diagonal are related with each measurable state and can be chosen arbitrarily or through intense testing.

$$R_k = \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \tag{8.14}$$

8.1.3 Filter Calibration

The filter will work best when the values that make up R_k and Q_k matrices are well calibrated. In order to optimise the choice of values of these matrices a set of tests can be made. These tests validate filter quality and convergence [48].

The filter performance can be evaluated with the innovation $\tilde{y}_k$ and its covariance S_k referring to each state. First of all, one should check if the innovation of each state is consistent with its covariance [48]. For this, the magnitude of the innovation should be about 95 % within the boundaries established by its covariance.That is given by:

$$\sigma = \pm 2\sqrt{S_k} \tag{8.15}$$

Figure 8.1 is an example of the magnitude representation innovation for both states x and y of the state space vector. Note that in both cases it is within the $\pm 2\sigma$ limits imposed

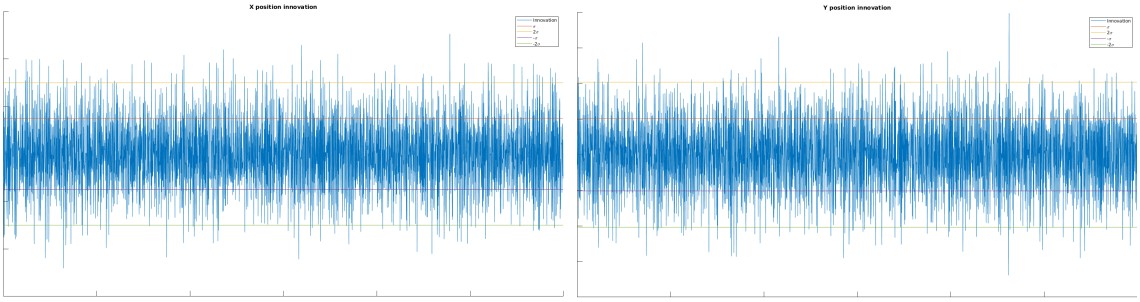


Figure 8.1: Innovation being tested with its covariance for both the states x (left) and y (right)

by Equation 8.15. According to [48] this test alone is sufficient to test the consistency of the implemented filter, however the remaining tests were performed.

The chi-square test would be responsible for verifying if the innovation was unbiased. First the normalised innovation, $q_{\tilde{y_k}}$, is calculated by Equation 8.16:

$$q_{\tilde{y}_k} = \tilde{y}_k \cdot S_k^{-1} \cdot \tilde{y}_k \tag{8.16}$$

With $q_{\tilde{y_k}}$ it is possible to calculate a moving average, with Equation 8.17. Moving average, $\overline{q}_{\tilde{y}_k}$, is an average that is updated over time, i, with new measurements, where N is the nth measure.

$$\overline{q}_{\tilde{y}_k} = \frac{1}{N} \sum_{i=1}^{N} q_{\tilde{y}_k}(i) \tag{8.17}$$

The set of values that make up the moving average must be between $r1$ and $r2$, in order to pass the test [48]. These values are the result of the formula represented in the form of Equation 8.18. where a sample space, N, of 100 was chosen alongside a 97.5 % confidence on the chi-square distribution. Knowing this, using the table in Appendix D and applying Equation 8.18:

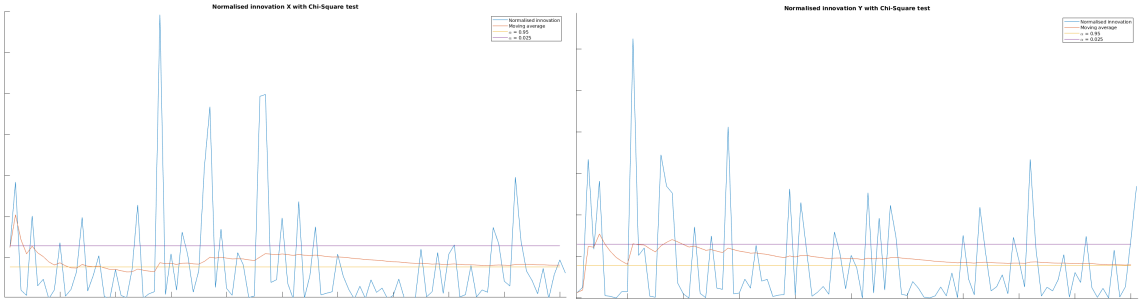


Figure 8.2: Normalised innovation with the chi-square test for both the states x (left) and y (right)

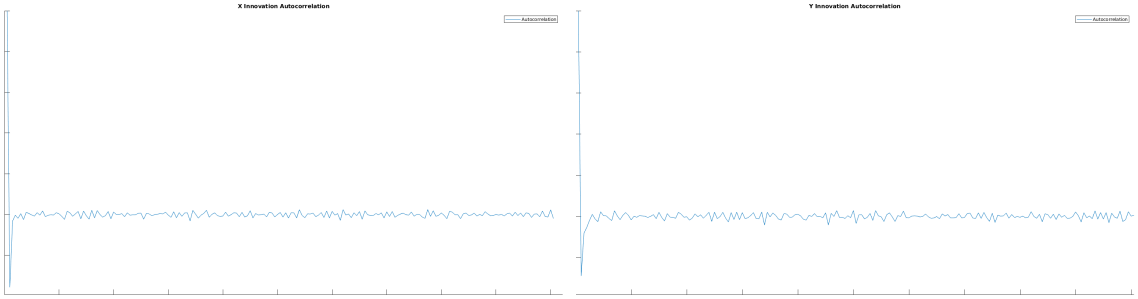


Figure 8.3: Innovation's autocorrelation of states x (left) and y (right)

$$[r_1, r_2] \;=\; \left[\chi^2_{100}(0.025)\;,\;\chi^2_{100}(0.975)\right] \;=\; [74.22,\;129.56] \tag{8.18}$$

Observing Figure 8.16 if it possible to confirm that the moving average results are indeed inside the imposed boundaries.

The last test would depend on the outcome of the innovation autocorrelation. What is expected is the existence of a peak at the first iteration and then the average resulting from autocorrelation should ideally be close to zero [48], which can be seen in Figure 8.3 for both states x and y.

The document [48] allows manual calibration of the necessary parameters for the proper functioning of the filter, explaining how to proceed according to the results of each of the tests presented above. All tests were successfully passed after some tweaks in matrices Q_k and R_k values.

A method was implemented to calibrate the values from the observation matrix, R_k, without manual calibration. Since the environment where the filter was applied was a simulation environment, several simulations were ran where the target would be stopped, assuming that the receiver was also stationary. Each simulation would recreate a mission with one hour and thirty simulations were generated from sratch in total with the target in a different position in each one of them. Knowing this, the average of each estimated position was computed and with it a standard deviation of all the samples from the simulations was computed.

8.2 Simulator and mission scenarios

Using the Matlab software, a simulator was generated to validate and test the DoA algorithm in scenarios closer to reality. Not only that, but also to enable the validation of two methods that would allow PE through data obtained via DoA.

The simulator has a set of parameters that can be changed through a Graphic User Interface (GUI) that was implemented so simulations are easier to run and more intuitive to control. The simulator's GUI can be seen in Figure 8.4.

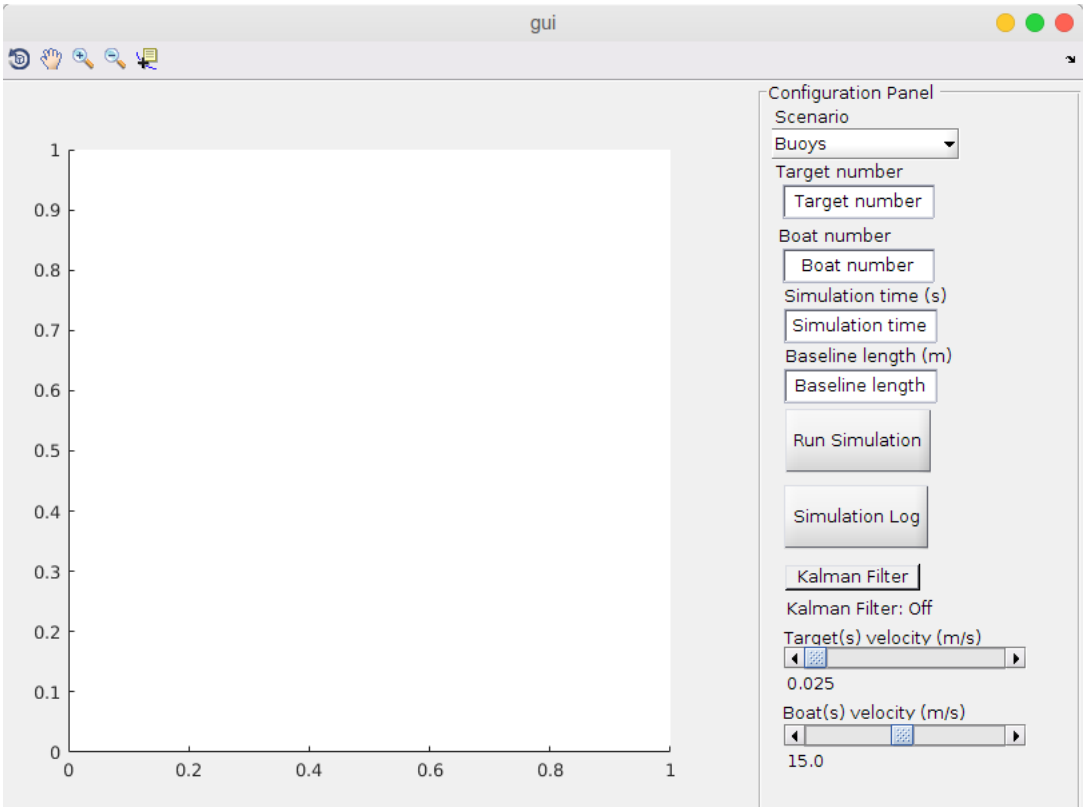


Figure 8.4: Simulator's GUI

The GUI allows the user to decide which mission scenario will be simulated, the number of targets that will be generated, and the number of boats that will follow them if the boat/ASV scenario is chosen. The user can choose how long the mission will be in seconds, as well as the size of the triangle side that forms the baseline of the acoustic reception system in meters. Three buttons can be pressed:

- Run Simulation - will start the simulation. Default values will be used if none of the fields are changed or if something is entered that makes no sense in the context, e.g a word in the "Target number" checkbox.
- Simulation Log - will reproduce the maximum, minimum and average errors of the estimated positions obtained from each boat/buoy in relation to each of the targets.
- *Kalman* Filter - This push button turns the filter On or Off before any simulation is started. When it is On, errors are added to the simulation to have results that are closer to reality and to test how the filter behaves.

Two scroll bars were also added so that the velocity of both targets and boats can be controlled. Finally the GUI will draw a graph containing the simulated boats/buoys as well as the targets to be followed. In addition the estimated positions are plotted. The

Table 8.1: Parameters not accessible through the simulator's GUI

Parameters	Units of Measurement
Simulation step	s
Target detectability rate	-
Angle limits	°
Filter initialisation with measurement	-
dt	s
A_k	-
H_k	-
P_0	-
R_k	-
Q_k	-

simulator contains a file that has all the necessary initialisations to function according to the user's decisions. However, some of those parameters may be changed and are not accessible through the GUI, being directly altered in the implemented code. Those parameters are listed in Table 8.1 and are not considered in the GUI for aesthetic reasons and for not being commonly modified. These parameters include the simulation step, target detectability rate, the limits of angles that the targets or boats can make in their movements (pitch and yaw) and the *Kalman* filter settings, including if it starts with the first measurement or not.

Two approaches were created in order to obtain an estimated position and consequent tracking with the computed DoA data. These approaches can be seen in the form of mission scenarios, one where two buoys would form a gate and other where an ASV would follow the target, orbiting it.

For both cases the trajectory of the target is calculated. For each of the targets, a new random position for the 3D plane is calculated by computing the new point at x, y and z. Positions are random to simulate a target whose movement is unpredictable. The target position is calculated with Equation 8.19:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} x_{i-1} \\ y_{i-1} \\ z_{i-1} \end{bmatrix} + \begin{bmatrix} v_{x_i} \\ v_{y_i} \\ v_{z_i} \end{bmatrix} \cdot s_{step} \tag{8.19}$$

Where x_i, y_i and z_i are the target's new position, x_{i-1}, y_{i-1} and z_{i-1} the previous position, *step* the simulation step and v_{x_i}, v_{y_i} and v_{z_i} the target's velocity in each coordinate. Target speed will vary over time on each axis. The simulation does not allow the speed value to reach the previously chosen limits. After calculating the velocity, Equation

8.19 is used compute the current target's position.

However, given the angle limit previously imposed on the simulation parameters, not all calculated points are accepted. If the pitch and yaw values exceed the limit imposed by this parameter, a new position must be calculated. This can be tested with the angle between two vectors, given by Equation 8.20:

$$cos\ \alpha\ =\ \frac{\overline{a}\cdot\ \overline{b}}{\overline{|a|}\cdot\ \overline{|b|}} \tag{8.20}$$

To compute Equation 8.20 three points are necessary. Depending on whether it is to calculate the pitch or yaw angle, the coordinates to use are different. For yaw Equation 8.21 is used:

$$\begin{cases} a\ =\ [x_{i-1}-x_{i-2},\quad y_{i-1}-y_{i-2}] \\ b\ =\ [x_i\ -\ x_{i-1},\quad y_i-y_{i-1}] \end{cases} \tag{8.21}$$

For pitch Equation 8.22:

$$\begin{cases} a\ =\ [y_{i-1}-y_{i-2},\quad z_{i-1}-z_{i-2}] \\ b\ =\ [y_i\ -\ y_{i-1},\quad z_i-z_{i-1}] \end{cases} \tag{8.22}$$

When the simulation is initialised the target starts with two positions already generated so this notion can be applied. With these restrictions the generated path movement, even if random, resembles a trajectory that would be possible to find in a real scenario. In Figure 8.5 it is possible to see how pitch and yaw vary over time, being left and right, respectively. A smooth trajectory was generated as was intended.

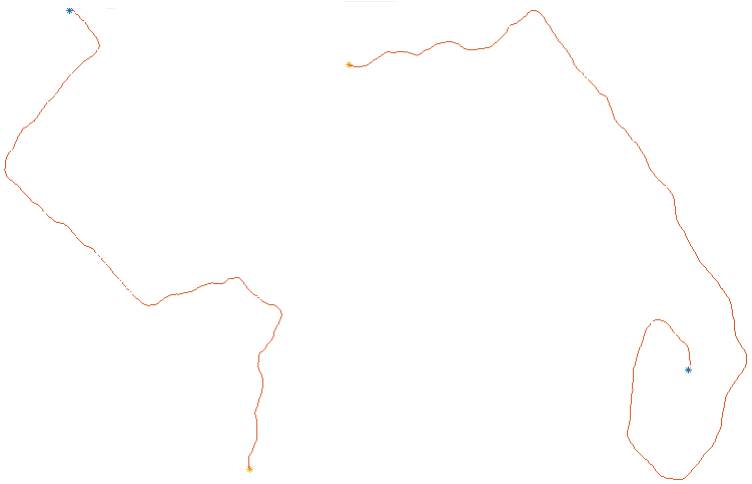


Figure 8.5: Example of a generated trajectory. Pitch,Y-Z view, on the left and Yaw, X-Y view, on the right

For both the scenarios the position is estimated through line intersection, using Equation 8.23. A line equation can be computed since the DoA is known, attaining the line slope through it.

$$\begin{cases} x &= \frac{(b2-b1)}{(m1-m2)} \\ y &= m1 \ * \frac{(b2-b1)}{(m1-m2)} + \ b1 \end{cases} \tag{8.23}$$

However, to have a line intersection at least two different DoA are needed and, preferably they shouldn't be parallel to each other, avoiding floating point errors in the simulation. With Equation 8.24 the angle between lines, θ, can be computed, avoiding aligned or parallel lines. If θ is bellow a chosen threshold, the position won't be estimated.

$$\Theta \ = \ |\frac{(m_2 \ - \ m_1)}{(1 + m_2 * m_1)}| \tag{8.24}$$

Where m_x are the slopes of each line.

For both solutions to work a baseline is created based on the simulator configuration parameters, namely the size of the baseline, i.e the size of the side of the triangle that makes up the receiving system. As a result, the distance to the centre of the equilateral triangle is calculated with Equation 8.25:

$$centre \ = \ \frac{\sqrt{3}}{3} \cdot l \tag{8.25}$$

Where *centre* corresponds to the triangle's centre and l to the triangle's side. Then, since the necessary condition for Algorithm 3 to work is equal angular distances, the baseline points are calculated by converting polar to Cartesian coordinates, using the set of Equations in 8.26. Using *centre* in relation to each of triangle's vertices and their angle in relation to the centre it it is possible to compute where each channel should be placed.

$$\begin{cases} x = \rho \cdot cos(\theta) \\ y = \rho \cdot sin(\theta) \end{cases} \tag{8.26}$$

Each channel of the receiving system corresponds to an angle. Since they are equidistant and three receivers are used, each of them is spaced 120 °. A rotation of 90 ° was

Table 8.2: Angle for each channel

Channel	Angle
1	0+90 °
2	240+90 °
3	120+90 °

made rotating the baseline making it more practical to use in the simulator. The angles of each channel are observable in Table 8.2.

For the buoys scenario an assumption is made: each buoy can communicate their computed data with one another while being time referenced. Since this assumption is made at every DoA computation two different values are obtained, one for each buoy at that iteration. This allows the line intersection to occur as soon as there is a simultaneous ID for the same target. The buoys will be displayed in the simulated world with an offset in the x axis minimising the number of scenarios where their computed DoA are aligned or parallel.

The main advantages of this method of operation are that the target's estimated position has low errors and it's able to describe its trajectory with ease. In addition it can start to compute a position as soon as there is an ID in both buoys. However, this scenario can't track targets that go above the hardware's detectability range. To solve that problem more gates needed to be added along the river or where the mission unfolds, at the expense of using more hardware.

For the boat/ASV scenario the boat's trajectory is also computed by the simulator. The trajectory is generated with Equation 8.19 without taking z into account. The main idea of this method/scenario is to have the boat describe an orbit around the target. This decision arises from the need to avoid parallel or aligned lines being generated on different time iterations, since only one receiving system is used on the boat and not a stereo system such as with the buoys. Since each iteration the boat changes its position, so that the baseline remains referenced to the boat it is necessary to make its rotation and translation in the world. For this Equation 8.27 is applied to each channel of the baseline at each iteration.

$$T = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} x_{i-1} \cdot cos(\theta) - y_{i-1} \cdot sin(\theta) + x_T \\ x_{i-1} \cdot sin(\theta) + y_{i-1} \cdot cos(\theta) + y_T \\ z_{i-1} + z_T \end{bmatrix} \tag{8.27}$$

Where T corresponds to the resulting transformation matrix, where the rotation, θ, followed by the translation ,$[x_T, y_T, z_T]$, is applied to the previous iteration, $i-1$, at each of the channel's coordinates, $[x_i, y_i, z_i]$. The translation is given by the boat's current position and the rotation by how its bearing varied from the past iteration to the current one.

Like with the target's trajectory some conditions are imposed, i.e impossibility of going underwater, making turns that wouldn't be possible (controlling the boat's yaw, applying Equation 8.21) and many others. After some DoA measurements while moving with a random trajectory to avoid false positives, it starts to track a previously selected target while describing an orbit trajectory. This trajectory ensures that parallel or aligned lines are avoided. This trajectory is possible by applying a normal between the boat's

bearing and the resulting DoA detection at each iteration. The more detections and/or the higher the target emission rate, the smoother it is the described orbit. So for estimating one position in this scenario it is mandatory that two detections are made at different locations. Instead of having a perfect orbit around the target with a normal between the boat's direction and the DoA, 3 ° are subtracted to the boat's bearing, increasing the proximity to the target. However, if the target's velocity increases so will the estimated position error, since for the same time interval the target's position is considerably different when compared with a slower target, that's why the target's trajectory isn't described as smoothly when compared with the buoys scenario, where the intersection is made at the same instant.

As with the DoA error analysis made in Chapter 7, the simulation estimated position errors can be calculated since the real target's position is known.

Chapter 9

Receiver and emission systems

As mentioned in Chapter 5 many improvements needed to be done in relation to [3] in order to meet the objectives of this project. The improvements that stand out the most are the new receiver system and an extra emission system that was made to allow the replication of tag signals and to make the needed tests easier, taking into account the slow emission rate that the V7 tags have. This chapter will contain the designed and developed hardware for the solution for both the emission and reception systems, while being validated through LTspice simulations. The software will also be explained in this chapter for both systems. The concepts for these systems will also be validated through oscilloscope prints, besides what was simulated. The real time software developed in Robot Operating System (ROS) will also take part of this chapter, which is a crucial for the solution integrate in a robot.

9.1 Receiver system hardware

As said in Chapter 5 the setup that was used in [3] didn't meet the requirements for the current project. The ZOOM UAC-2 soundboard had a sampling rate of 192 kHz and the gain could be changed manually. However, the sampling frequency was barely above the Nyquist frequency (in relation to the tag's signal) and it depended on the manufacturer's software, which wouldn't give any information about the received voltages of the pair PA-4 preamplifier and the AS-1 hydrophone. Not only that, but the data could only be logged essentially through an audio recording software. To have more control of the hardware's inputs and outputs and to enable near to real-time solutions with higher sampling rate, developing the new receiver system was a necessity.

The new receiver system is composed of:

- One AS-1 Hydrophone from Aquarian Audio
- An EF125 high-pass filter from Thorlabs

- One male-to-male Bayonet NeillConcelman (BNC) connector
- The NUCLEO-H743ZI from STMicroelectronics
- A shield board on top of the uC for signal conditioning
- A power board that stacks on the signal conditioning board

The AS-1 hydrophone is designed to be sensitive and with a linear response between 1 Hz to 100 kHz which covers the tag's signal frequency. Aquarian Audio claims that this hydrophone is omnidirectional while receiving on the horizontal axis at all frequencies and that it can resist to a pressure that is equivalent to a depth of 30 m, for at least two hours [49].

To minimise the noise that would be felt during the data logging of acoustic signals the EF125 is used. It is an in-line passive high-pass filter from Thorlabs with a passband window that starts roughly at 50 kHz. The band-rejection cuts off frequencies which are prominent from the underwater environment especially the human audible frequencies that range between 20 Hz to 20 kHz. Figure 9.1 shows a relation between the frequencies and the relative response in dB to those frequencies [50]. Both the filter's input and output are female BNC and to connect it to the signal conditioning board, which also had a female BNC connector for its input, the male-to-male BNC connector was used.

The NUCLEO-H743ZI evaluation board was used, since with this board it was easier to use the STM32H743ZI uC. The uC belongs to the Advanced RISC Machine (ARM) architecture family and has a maximum working frequency of 400 MHz. This board with that uC is one of the STMicroelectronics high-end evaluation boards and its features meet the solution requirements namely the Ethernet modem (which allows the transmission of datagrams through ethernet) and three analog-to-digital converters (ADC)s that can be configured up until 16-bit . At that resolution the ADC can sample at a maximum

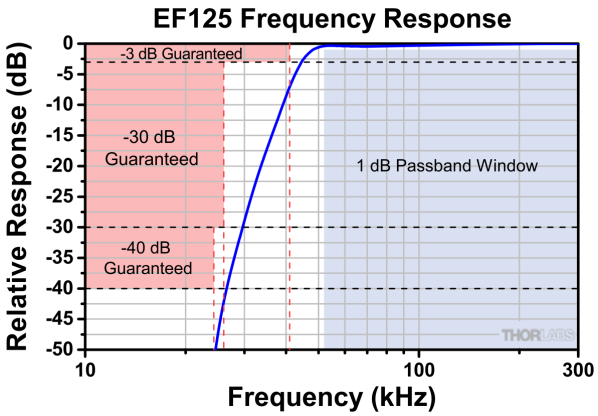


Figure 9.1: EF125 frequency response [50]

frequency of 3.6 MHz [51]. This uC can accept in its ADC signals with voltage values between 0 V and 3.3 V. Audio signals are usually centred at 0 V, which by itself would be a problem to be directly received through the uC ADC. Not only that but the expected signals will usually have voltages in the mV range. After some tests the maximum obtained voltage, under extreme conditions, i.e the hydrophone smashing to a surface or a strong acoustic source right next to it, was in the order of ± 4 V, which would be above of the maximum accepted by the uC.

9.1.1 Signal conditioning board

The signal conditioning board was made to make the voltage levels acceptable for the uC, protecting the latter. Not only that but also for amplifying the medium received signals. The circuit needed to centre the received audio signals in the middle of the voltage range that the uC can accept, so for that a direct current (DC) offset needed to be added. Ideally the mean value of the signal should be around 1.6 V. Since analog signals are dealt by the circuit the noise was a main issue to take into account.

Three operational amplifier (Opamp) circuits were used. For the first circuit a non-inverting Opamp configuration was applied, as the one shown in Figure 9.2. In this configuration the input signal, V_{in}, is routed through the Opamp's positive input, V_+. Assuming that the Opamp has high impedance inputs, the assumption of an ideal summing point can be made:

$$V_+ \approx V_- \tag{9.1}$$

And that:

$$V_{out} = R_1 * I_1 + R_2 * I_2 \ , \ I_1 = I_2 = \frac{V_-}{R_1} \tag{9.2}$$

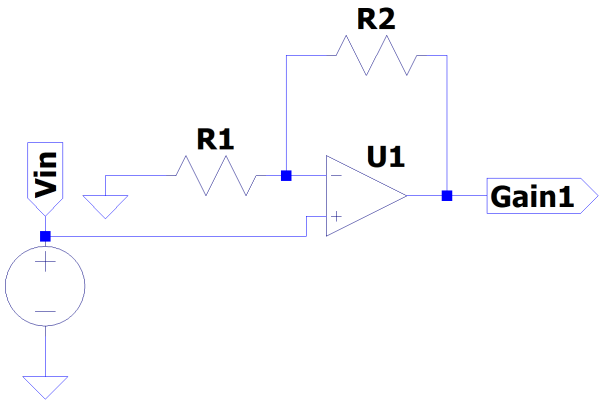


Figure 9.2: Non-inverting Opamp configuration

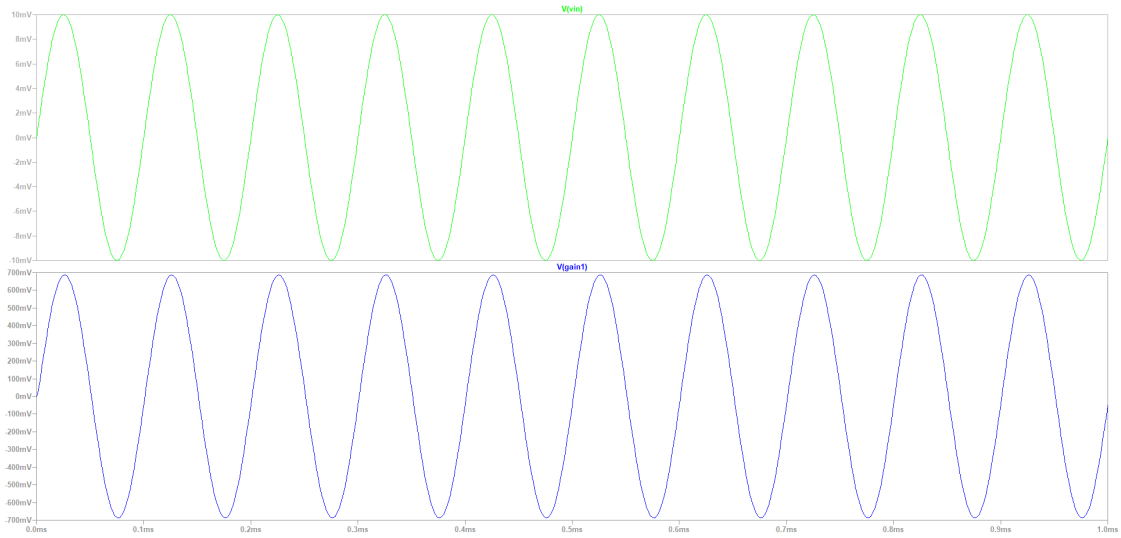


Figure 9.3: Input (green/up) and output (blue/down) of the non-inverting Opamp configuration

Where V_- is the Opamp's negative input, R_x the used resistors and I_x the current that passes through them. With Equations 9.1 and 9.2, Equation 9.3 is obtained:

$$V_{out} = V_- * \frac{R_1 + R_2}{R_1} \tag{9.3}$$

With equations 9.1 and 9.3, the output of a non-inverting Opamp configuration can be given by 9.4:

$$V_{out} = V_{in} * \frac{R_1 + R_2}{R_1} = V_{in} * \left(1 + \frac{R_2}{R_1}\right) \tag{9.4}$$

Where the voltage gain, $A_{(V)}$ is given by:

$$A_{(V)} = \frac{V_{out}}{V_{in}} = \left(1 + \frac{R_2}{R_1}\right) \tag{9.5}$$

It was easy to change the $A_{(V)}$ of this configuration in the final Printed Circuit Board (PCB), which allowed to test which values the resistors R_1 and R_2 should had for optimal results. The R_1 got fixed at 10 kΩ whereas the R_2 value changed according to tests made in the tank with a V7 tag, which had a signal amplitude of approximately 0.01 V when close to the setup. The starting value for R_2 was 2.2 MΩ, which would saturate the voltage accepted by the Opamps, given the voltage that powered them. This wasn't ideal since information would get lost and not read by the ADC. After some tests 680 kΩ was the chosen optimal value for R_2. Taking into account Equation 9.5 this circuit's gain is 69. A simulation was made and the results are observable in Figure 9.3 where the x axis is the simulation time and the y axis represents the signal voltage at a given instant. The

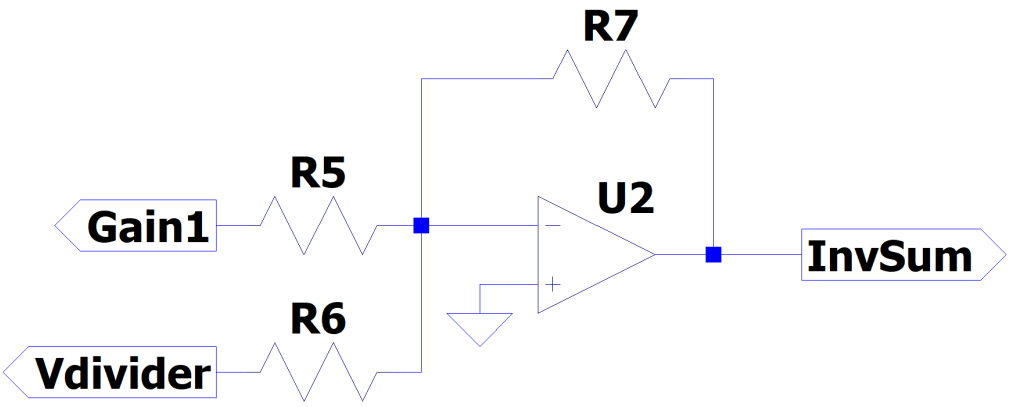


Figure 9.4: Summing Opamp Configuration

simulation used as an input a sine wave with 0.01 V, as the V7 tag, at a frequency of 10 kHz (which wasn't important for the simulation, since only the voltages were relevant for the signal conditioning). The input is shown as green and the output as blue, represented as Gain1. The output signal has the same frequency as the input signal and a voltage of 690 mV as was expected.

In order to add the DC offset the summing amplifier configuration was used, as the one shown in Figure 9.4. This configuration is also called in literature as summing inverter, since the output signal is inverted when compared with the input signal.

Theoretically according to the Opamps properties the input resistance of an ideal operational amplifier is near to infinite, thus the next assumption can be made:

$$I_7 \;=\; I_5 \;+\; I_6 \tag{9.6}$$

Since the input resistance is near to infinite and using Equation 9.1 the voltage at both the positive and negative inputs of the Opamp will be 0 V. By applying the Ohm's law to Equation 9.6, Equation 9.7 is deducted:

$$-\frac{V_{out}}{R_7} \;=\; \frac{V_5}{R_5} \;+\; \frac{V_6}{R_6} \tag{9.7}$$

If the deduction is made in order to V_{out}:

$$V_{out} \;=\; -R_7 \;*\; \left(\frac{V_5}{R_5} \;+\; \frac{V_6}{R_6}\right) \tag{9.8}$$

If the input resistors have equal values, $R_5 = R_6 = R_{in}$, then:

$$V_{out} \;=\; -\frac{R_7}{R_{in}} \;*\; (V_5 \;+\; V_6) \tag{9.9}$$

Where the voltage gain for this configuration is given by:

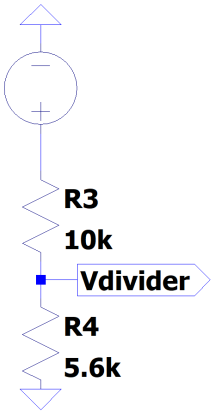


Figure 9.5: Voltage divider

$$A_{(V)} = -\frac{R_7}{R_{in}} \tag{9.10}$$

In case all the resistors in the configuration are the same, then only the DC offset is added to the input signal, as shown in Equation 9.11:

$$V_{out} \;=\; -(V_5 \;+\; V_6) \tag{9.11}$$

As mentioned in the beginning of this subsection, ideally the signal's voltage should be centred at 1.6 V, half of the ADC's voltage range. No amplification was applied in this circuit, being only applied the DC offset to the output of the first configuration, V_{gain}. To do this, Equation 9.11 was applied, knowing that all the resistors needed to be the same,

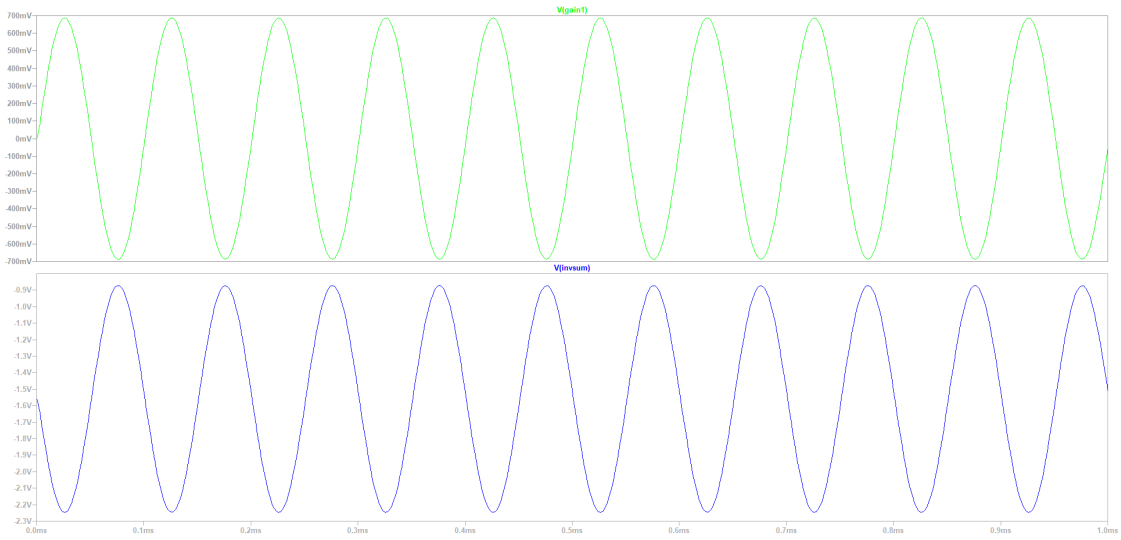


Figure 9.6: Input (green) and output (blue) of the Summing Opamp configuration

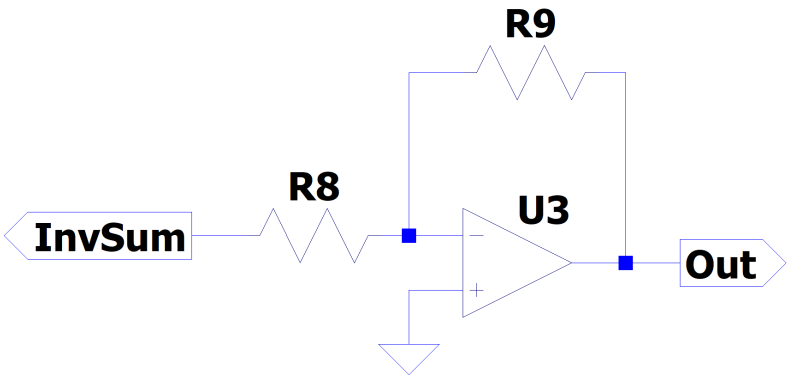


Figure 9.7: Inverting Opamp configuration

resulting in a DC offset of -1.6 V to to the original signal, resulting in the output Inv_{sum}. To apply the voltage V_6 to generate the offset a voltage divider was used, as the one shown in Figure 9.5. The voltage that powers the PCB is yet to be explained in this subsection, but ideally the circuit's input voltage,V_{power_+}, should be around 5 V. Knowing this, the voltage divider equation was applied:

$$V_{divider} = \frac{V_{power_+} \ * \ R_4}{(R_3 \ + \ R_4)} \tag{9.12}$$

This configuration was simulated, having the results shown in Figure 9.6. The green graph on top is this circuit's input and the bottom one is the summing Opamp configuration result with the above conditions. The resulting signal is the same as V_{gain1} but centred at approximately -1.6 V, $V_{inv_{sum}}$

Finally the signal needed to be inverted, in order to only have positive inputs which are already centred at 1.6 V. For that a inverting Opamp configuration was used, replicated in Figure 9.7.

The input signal,$V_{inv_{sum}}$, is connected to the the Opamp's negative input, V_- and the positive Opamp input is connected to the ground. Assuming the Opamp high impedance property and applying Equation 9.1, then:

$$\begin{cases} I_8 \ = \frac{V_{inv_{sum}}}{R_8} - \frac{V_-}{R_8} \\ I_9 \ = \ \frac{V_-}{R_9} \ - \frac{V_{out}}{R_9} \\ I_8 \ = \ I_9 \end{cases} = \begin{cases} I_8 \ = \frac{V_{inv_{sum}}}{R_8} \\ I_9 \ = \ -\frac{V_{out}}{R_9} \\ V_{out} \ = -\frac{R_9}{R_8} \ * \ V_{inv_{sum}} \end{cases} \tag{9.13}$$

Where the voltage gain is given by:

$$A_{(V)} = -\frac{R_9}{R_8} \tag{9.14}$$

However the signal only needed to be inverted, so the resistors values where the same to not add gain to the output signal, V_{out}. This configuration was also simulated, obtaining

the results seen in Figure 9.8. In the top represented as green is the input of the third and last circuit, the result of the summing circuit $V_{inv_{sum}}$. This signal was inverted resulting in the bottom graph, represented in blue, V_{out}. The signal is conditioned as was proposed in the beginning of the subsection. The signal is amplified while being centred at 1.6 V at the middle of the ADC acceptance voltage range. For the uC's protection a Zener of 3.3 V was used in the circuit's output. One 0 Ω resistor (jumper) was used to allow the PCB's users to decide if the signals output would go directly to the ADC or to the uC's internal Opamp.

The chosen Opamps for the PCB were the OPA2192 rail-to-rail Opamps from Texas Instruments. This is a high precision Opamp which is necessary for the solution to avoid noise in the ADC readings as the signal passes through them. It also has a low offset voltage and having rail-to-rail input and output is ideal, since it allows signals for both the input and output to reach until the voltage supply accepted by the Opamp [52]. Since the PCB's input signal is centred in 0 V, then both positive and negative values needed to be accepted in the circuit. The OPA2192 can be powered for both positive and negative voltage supplies until ± 20 V. As mentioned in this section, the PCB would be stacked in the uC where it would get its supply voltage. However the uC only can only power other devices with its 5 V voltage supply. Both the positive and negative voltages are needed for the circuit to work properly, in order to avoid losing data. To supply the necessary voltage a power board needed to be developed.

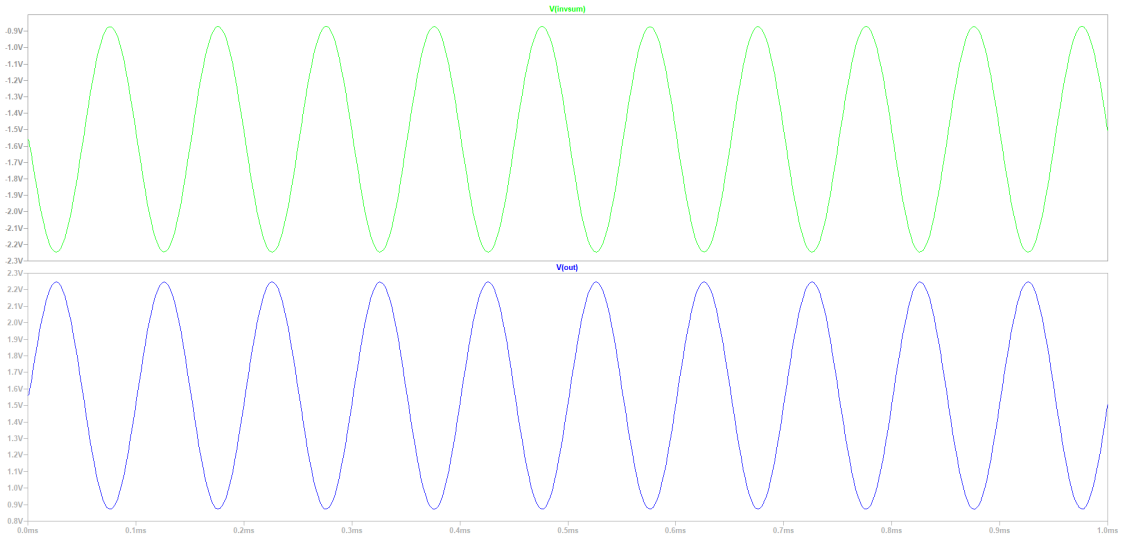


Figure 9.8: Input (green/up) and output (blue/down) of the inverting Opamp configuration

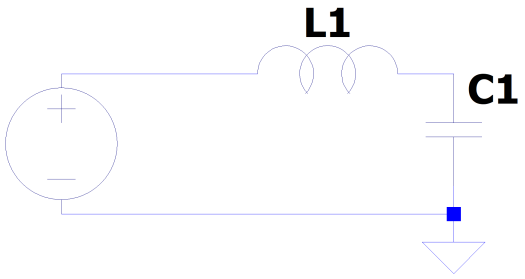


Figure 9.9: LC low-pass filter circuit

9.1.2 Power board

The uC 5 V DC voltage needed to be converted to a bipolar voltage, having both the negative and positive voltage output. Initially the MAX680 charge-pump integrated circuit (IC) from Maxim was used [53]. A charge-pump is a DC to DC converter which uses capacitors for energetic charges, working as a switching device. However, knowing that the low noise was an important requirement to be met, this IC was discarded, since it introduced noise in the Opamps through their voltage supply. The used IC was the LM27762 from Texas Instruments. This IC outputs a low-noise output which can be both positive and negative. The negative voltage is generated using a charge pump, like MAX680, but then it is followed by a low-noise negative low-dropout regulator (LDO). The positive voltage is generated from the input with a low-noise positive LDO [54].

For both the input and output of this IC a low-pass filter was used, as the one in Figure 9.9. That resonant circuit was used for both the IC's positive and negative output to also minimise any noise, even though there's a LDO for both the outputs. The LC circuit cutoff frequency, f_c, is given by Equation 9.15:

$$f_c = \frac{1}{2\pi\sqrt{LC}} \tag{9.15}$$

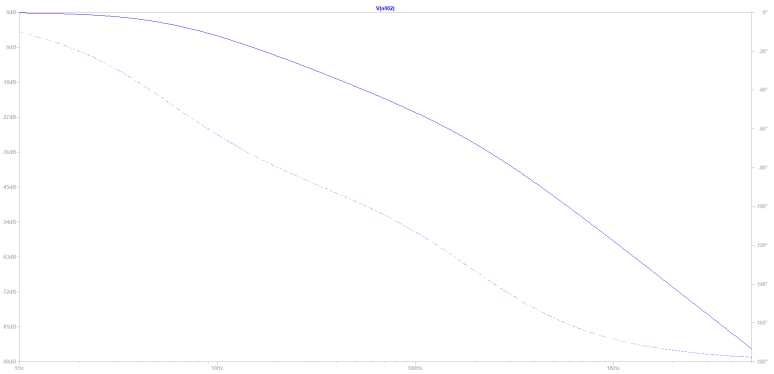


Figure 9.10: LC low-pass filter frequency response

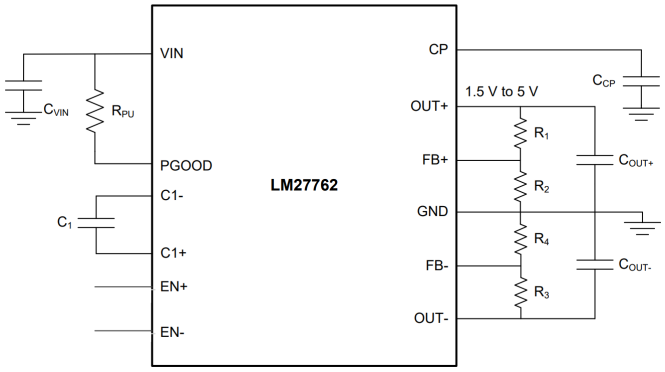


Figure 9.11: LM27762 implemented application note

Where L is the inductor's inductance and the C the capacitor's capacitance. In order to have a signal with low-noise a low f_c value tried to be attained. For that the chosen values for the inductor and capacitor were 10 mH and 2.2 mF, respectively. With those values the frequency response is as it shows in Figure 9.10, where the f_c value is approximately 34 Hz.

To have the $\pm$ 5 V output for the Opamps' supply voltage the recommended values from WEBENCH Power Designer from Texas Instruments were used as well as the application note seen in Figure 9.11. The board layout can be found in Appendix B.

The whole receiver system can be seen in Figure 9.12 and the receiver schematic can be seen in Appendix A with the power board and the LC circuits.

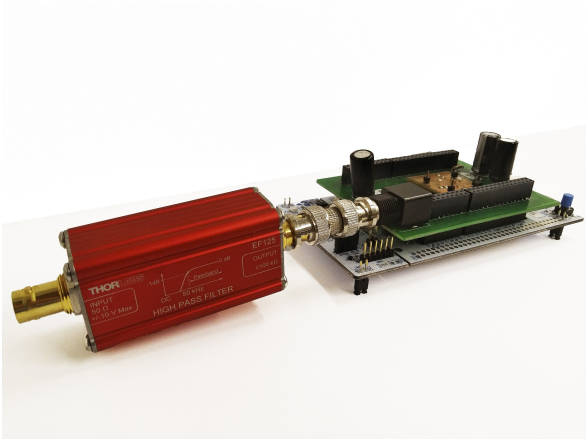


Figure 9.12: Receiver system

9.2 Emission system hardware

The emission board can be seen as an extra system for the project since it allows faster emissions of replicated V7 tags signals. This allowed for more data to be studied in less time.

The emission system is composed of:

- One AS-1 Hydrophone from Aquarian Audio
- The NUCLEO-F446RE from STMicroelectronics
- A shield board on top of the uC for signal conditioning

The AS-1 hydrophone besides the characteristics it has while receiving it is also omnidirectional when sending signals. It is capable of sending signals with a maximum 30 Vpp [49]. The NUCLEO-F446RE is also an evaluation board from STMicroeletctronics that was already used in the laboratory. The most important feature to take advantage of this board is its two 12-bit digital-to-analog converters (DACs), which would enable the creation of signals. Not only that but this evaluation board contains a uC that belongs to the ARM family that works at a maximum frequency of 180 MHz [55]. As with the receiver system the uC has a voltage range between 0 V to 3.3 V. The DAC's maximum output is comprised of that interval. If a sine wave is generated it will be centred in the middle of that interval, not replecating an expected audio signal. To enjoy the AS-1's maximum working voltages and to also strive for having the signal's maximum emission range, a conditioning signal board needed to be made.

The PCB strives for not only amplifying the DAC's generated signal but also centre it in 0 V. To do that a differential amplifier Opamp configuration was used, as shown in Figure 9.13. The same Opamp that was used in the receiver system was used in this system because of the characteristics already mentioned in the past Section.

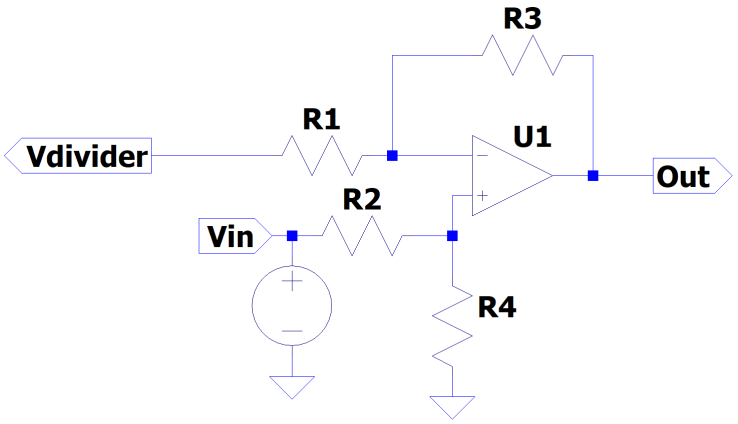


Figure 9.13: Differential Amplifier Configuration

Considering the currents that pass through the resistors and by applying Ohm's law:

$$I_1 = \frac{V_{divider} - V_-}{R_1} \ , \ I_2 = \frac{V_{in} - V_+}{R_2} \ , \ I_3 = \frac{V_- - V_{out}}{R3} \ , \ I_4 = \frac{V_+}{R_4} \tag{9.16}$$

Where $V_{divider}$ results from a voltage divider as the one in Figure 9.5, V_- and V_+ the negative and positive inputs of the Opamp, the I_x the current of the correspondent resistor R_x, and V_{in} and V_{out} the circuit's input and output respectively.

Using the the expressions $I_2 \approx I_4$ in order to V_+, since the Opamps have a high impedance in their inputs, from Equation 9.16 results:

$$V_+ = V_{in} \frac{R_4}{R3 + R_4} \tag{9.17}$$

Doing the same to the currents I_1 and I_3, in order to V_-:

$$V_- = V_{in} * \frac{V_{divider} R_3 + V_{out} R_1}{R_1 + R_3} \tag{9.18}$$

Knowing the Opamp's summing point constraint, Equation 9.1, then Equations 9.17 and 9.18 are equal to one another, resulting in the Equation 9.19, in order to V_{out}:

$$V_{out} = V_{in} * R_4 * \frac{(R_1 + R_3)}{(R_1 * (R_2 + R_4))} \tag{9.19}$$

If $R_1 = R_2$ and $R_3 = R_4$, V_{out} can be computed:

$$V_{out} = \frac{R_3}{R_1} * (V_{in} - V_{divider}) \tag{9.20}$$

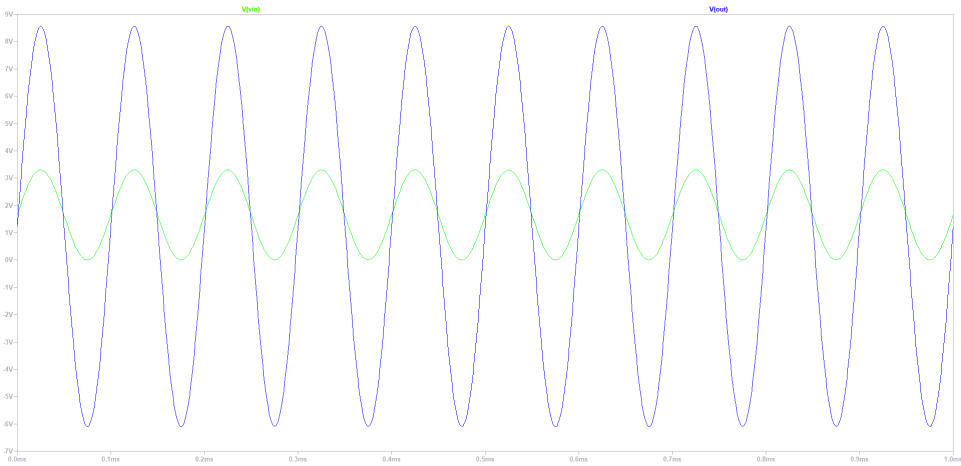


Figure 9.14: Input (green/up) and output (blue/down) of the Differential Opamp configuration

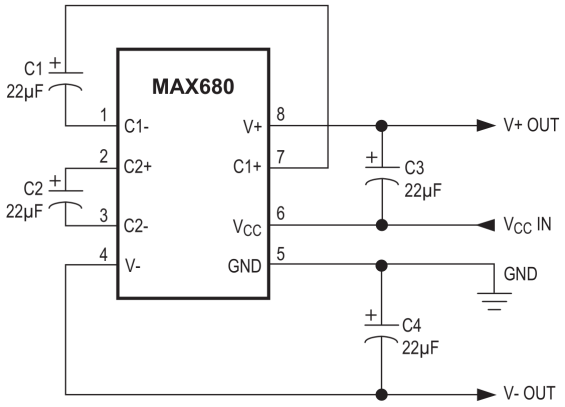


Figure 9.15: MAX680 application note's circuit [53]

To remove the uC DAC'S DC offset, approximately 1.65 V, $V_{divider}$ needed to have that value, according to Equation 9.20. For that Equation 9.12 is used. As said before, a gain was also applied to the signal. This configuration's gain is given by:

$$A_{(V)} = \frac{R_3}{R_1} \tag{9.21}$$

To have the desired $V_{divider}$, the chosen values were $R_1 = R_2 = 10$ kΩ and $R_3 = R_4 = 56$ kΩ. With those values the resulting $A_{(V)}$ is 5.6. A simulation was run, as shown in Figure 9.14. The simulation time is referenced through the x axis and y axis has the signal voltage V to each of those instants. The circuit's input represented as green is the DAC's simulated output, with the voltage mean at 1.65 V and 3 Vpp. The signal's output, represented as blue, with the simulated configuration, has 17 Vpp and a mean of 0 V.

To supply the PCB, as with the receiver system, the uC suply voltage is used alongside

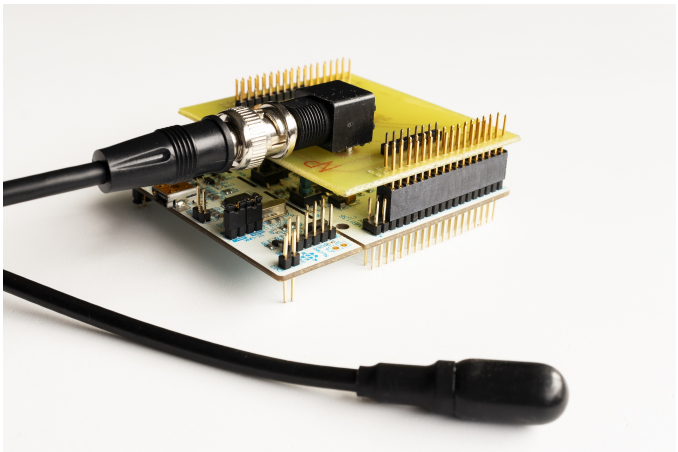


Figure 9.16: Emission system

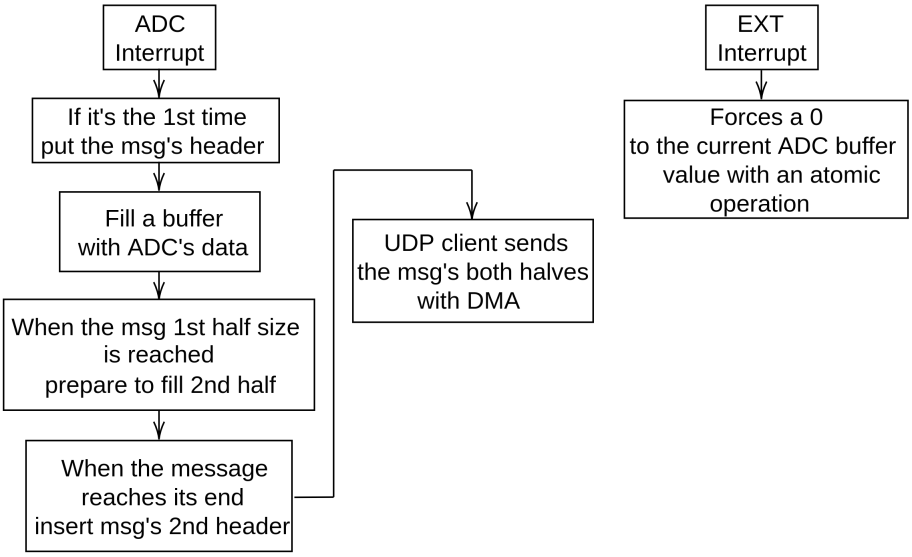


Figure 9.17: Receiver uC's high level Software

an IC. In this case the MAX680. It is also a charge-pump DC to DC converter, which supplies, given the manufacturer's application note shown in Figure 9.15, with a supply voltage of 5 V an output of ± 10 V [53]. This output is more than enough to allow the Opamp to have the 17 Vpp with a mean of 0 V output.

The whole emission system can be seen in Figure 9.16. The PCB's schematic is in Appendix C.

9.3 Receiver system Software

First of all a synchronisation trigger signal was created with the NUCLEO-F429ZI evaluation board and the STM32F4 HAL standard libraries both from STMicroelectronics. For that an output compare trigger with a configured pulse width modulation (PWM) method was used. The PWM was configured to have high polarity, in other words, the PWM's duty cycle is going to be the amount of time its output state is high. The configuration assured that the PWM duty cycle would be almost 100 %. The low state of the PWM's output would correspond to the trigger and would occur at each second. The receiver uC is also programmed with the STM32F4 HAL standard libraries alongside with STM32CubeMX tool, the last being useful to easily program the uC's configurations. The used peripherals were the ADC, the Direct Memory Access (DMA), the Ethernet output and a timer. The ADC was configured to acquire data at a rate of 500000 samples per second. The ADC stores the data in halves for it to comply with the used protocol to send the data via ethernet. User Datagram Protocol (UDP) was the chosen ethernet Protocol and it was programmed with the lightweight IP (lwIP) open-source TCP/IP stack which

is designed for embedded systems, while being used with DMA. Each of the halves is accompanied by a header with the corresponding time of recording that is given by the programmed timer. A pin is programmed to be triggered by an external interrupt that will be received at each second. The external interrupt will occur with a falling edge trigger, because of the way the trigger was programmed in the NUCLEO-F429ZI evaluation board. This is done to synchronise all the receiver boards which are triggered at the same instant. When this external interruption triggers the uC the current position in the ADC buffer will contain the value 0. The ADC will always be forced by the software to be above 0 to avoid false positives. Filling the ADC's buffer is always an atomic operation avoiding memory issues. Each receiver modem will have a specific IP address and will connect to an ethernet switch. The network configuration can be seen in Figure 9.18.

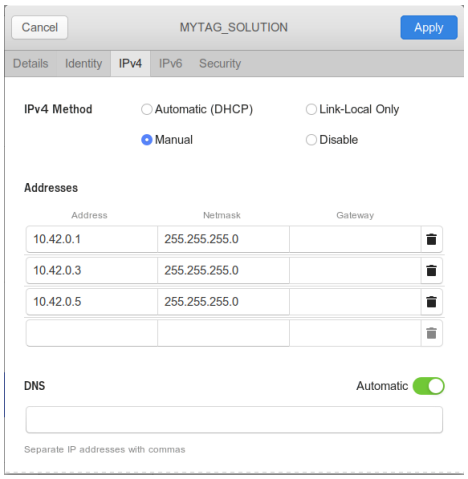


Figure 9.18: Configured Network

9.4 Emission system Software

This system was programmed with the STM32F4 HAL standard libraries from STMicroelectronics. The configured peripherals were an Universal Synchronous Asynchronous Receiver Transmitter (USART), a DAC and two timers. The code would only use interruptions, without any functions being called in the main function's loop. One timer is responsible for the intervals between pings and the other for the interval between signal emissions. The first one uses a buffer with the selected signal which contains the intervals between emissions. This timer is used by the function *pingSend* which is responsible for using the DAC peripheral to send the signal's pings. When an interval isn't reached the DAC value will be centred in its range, in this case around 2000, since it has 12-bits.

The second timer is used to establish the interval between emissions using a counter that increments at each interrupt. When the counter reaches the established value for the

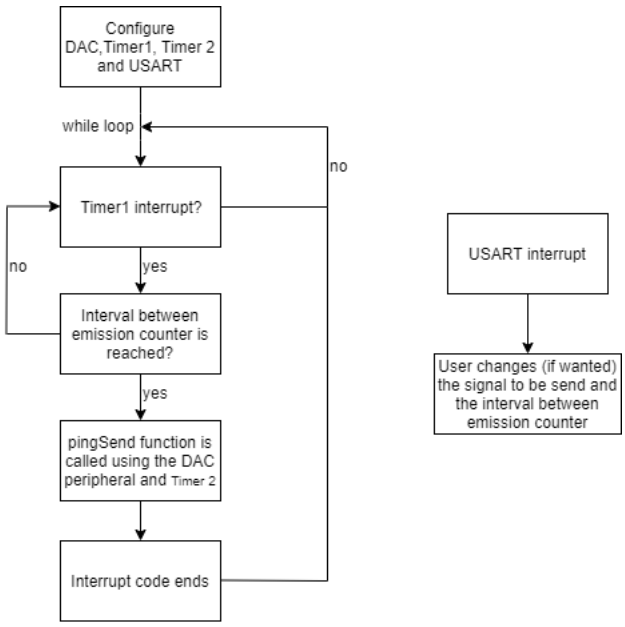


Figure 9.19: Emitter uC's high level Software

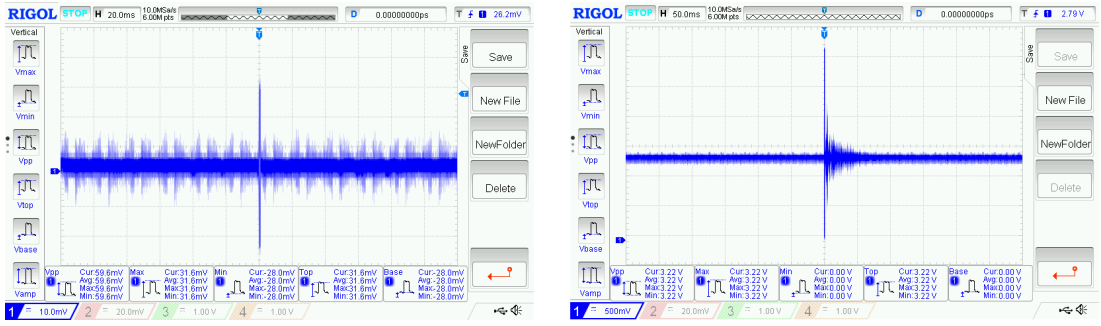


Figure 9.20: Ping in the AS-1 hydrophone's output (Left) and receiver's signal conditioning board output (right)

signal emission the function *pingSend* is called in this interruption. The interval between emissions can be selected via USART, not only that but also the signal to be send, if it is available in the uC's memory. The code is summarised in Figure's 9.19 diagram. There Timer1 corresponds to the one that controls the interval between signal emissions and Timer2 the one that is responsible of counting the time between pings.

9.5 Receiver and emission system tests

More tests needed to be done besides the simulations elaborated with Ltspice, to further the validation of the developed systems. Oscilloscope prints were gathered to observe the behaviour of the implemented software and the outputs of the designed hardware.

The conditioning signal PCB developed for the receiver system was tested through comparing its output, on the right of Figure 9.20, with that of the AS-1 hydrophone, on the left of Figure 9.20. The observable signal corresponds to a V7 tag ping. It is noticeable that the signal was amplified, from 59.6 mV to 3.22 V, corresponding to a gain of approximately 54.

The signal's average is 1.55 V according to Figure 9.21.

The powerboard circuit was also tested, having satisfactory results on both outputs, Figure 9.22. However the ± 5 V weren't attained, which affected the rest of the conditioning circuit, having for the positive output an average of 4.57 V and for the negative -4.45 V.

The interval between emissions was also acquired to validate if the programmed 1 s between emissions was correct. Figure 9.23 proves that the software was well implemented, since the interval was 1.060 s.

The DAC's output is observable on the left of Figure 9.24, where the tag 1308 was replicated successfully, validating the implemented software on the emission system uC. The conditioning signal board output is on the right of Figure 9.24 where the signal's average is approximately 0 V, (the vertical Oscilloscope reference is with an unintended offset) and with 17.4 Vpp.

The systems have been validated for use with the DoA algorithm.

9.6 ROS implementation - Real time solution

One of the main goals in [3] was to guarantee that the tags were detected and identified in real-time. Because of that the detection and ID algorithm proposed in [3] was adapted to be used with ROS. ROS is a robotics middleware with many software frameworks called packages that are available for anyone who uses it [56]. In ROS, packages refer to open source implementations of algorithms and drivers that are commonly used in robotics. Some default packages are included already when ROS distributions are installed.

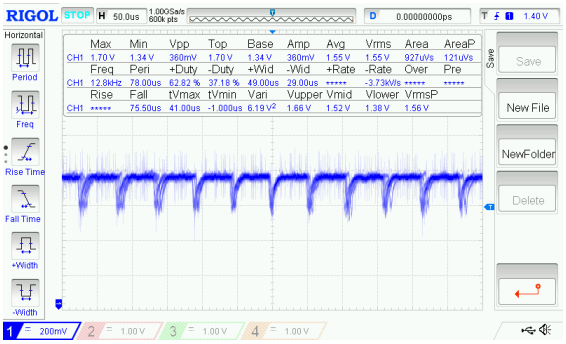


Figure 9.21: Receiver's signal conditioning board average voltage

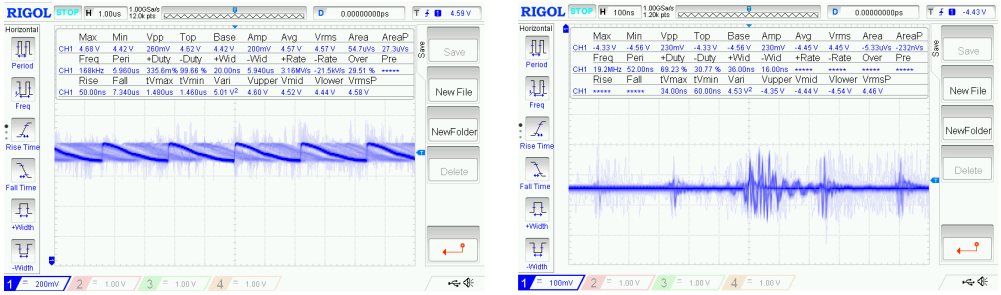


Figure 9.22: Powerboard positive output (left) and negative output (right)

Packages can be developed by anyone, as was the case in this solution, and shared through code distributing websites. Some important definitions need to be established so the explanation of what was implemented can be understood. ROS uses a graph architecture where processing takes place in nodes [56]. ROS needs to be booted with roscore, which provides the connection information for nodes to communicate with one another. Every node connects to the roscore as they are started. A node represents a single process in the ROS graph and each has a name that is registered to ROS master (roscore). Nodes are basically where most of the client code is made and are the ones which take actions based on information received form other nodes and can also send data to them [56]. Every ROS system needs a running roscore, otherwise nodes couldn't find other nodes [56]. For nodes to send and receive messages they need to either publish or subscribe to a topic, respectively. Topics are named buses for those purposes and their names must be unique. The messages' contents are defined by the programmer and usually there are standard messages in ROS that are useful to use when working with sensor data, state information and many others [56]. Very often a set of nodes is made to be launched automatically,

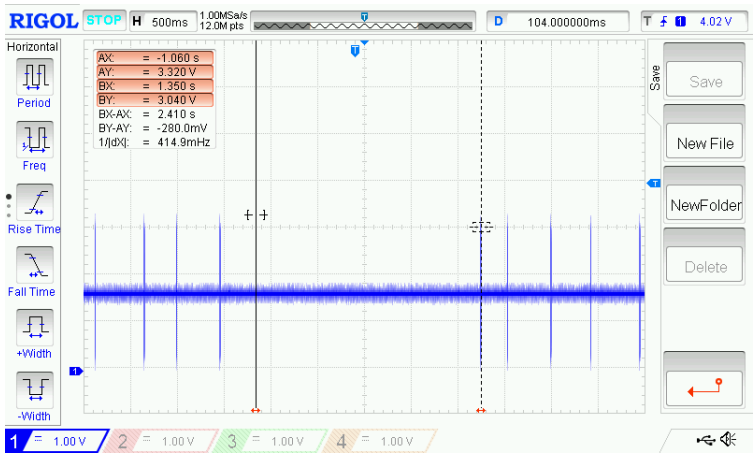


Figure 9.23: Interval between emissions software test

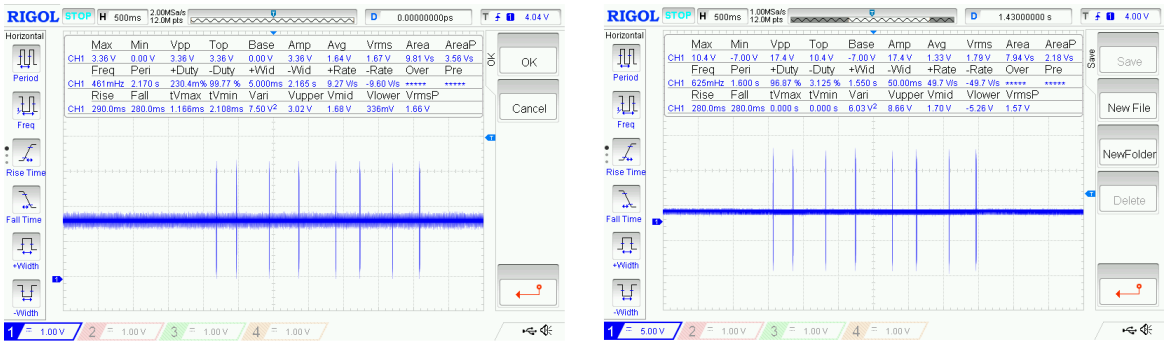


Figure 9.24: *u*C's DAC output (left) and emission's conditioning signal board output (right)

using the roslaunch tool.

The ROS implemented software is summarised in Figure 9.25. All the ROS nodes were implemented in Python.

As can be seen in Figure 9.25 all the processing software was implemented as ROS nodes to allow the DoA computation to be made in real time. In Figure 9.25 the x refers to one of the three acoustic channels. The UDP datagrams are collected in the node *ad_receiver_x*, which checks which half of the sent datagrams was received and sends it as joy standard message from ROS. After that the data is published through the topic *ad_measurements_x*.

The node *Message_listener_x* will subscribe to that topic. This node is responsible of accumulating samples from ADC measurements to create a window which will be used in the *TagFinder_x* node. In addition to that it queues measurements to be used in future iterations, avoiding data loss. The node waits for at least 6 datagrams of data to be received, where each byte corresponds to a sample. This node publishes the data

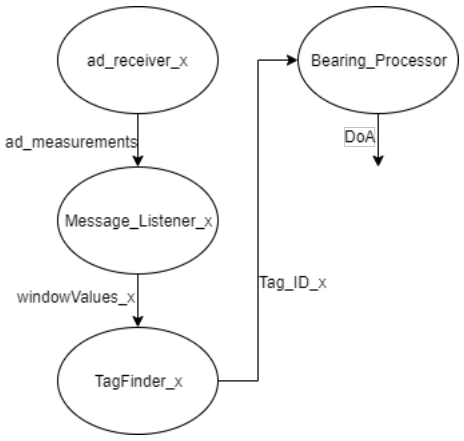


Figure 9.25: Implemented ROS nodes software diagram

with the Float64MultiArray ROS standard message, with a topic named *window_values*. Each message contains a window with 3072 samples, corresponding, approximately, to the duration of a V7 tag ping.

The *TagFinder_x* node is an adaptation of the detection and ID algorithm that was presented in [3]. This node uses a sliding window method with a duration of 4 s for the detection and ID algorithm to be applied. By using this technique and optimised libraries this process is achieved in real-time. Before using the cross-correlation, as mentioned in [3], the software fills an array with the samples that correspond to the 0 values that were sent by the ADC. Those samples coincide with the instant that the synchronisation triggers were sent. After detecting the triggers, those ADC values are switched with the previous value or the next in case it was the first in the data buffer. The software not only guarantees that no data is lost, saving detected pings for posterior analysis, but also uses a portion of the previous window, avoiding losing a ping in a window transition. Those variables are cleared when enough time as passed where no ID could be made. Memory is something to be taken into account when dealing with so many samples, so the software clears up its variables publishing only the essential. The detected tags are published in the form of the topic *Tag_ID_x*. In addition to publishing the tag's ID and its detected pings, it also contains the two nearest synchronisation samples related with the first ping of the detected tag.

The *Bearing_Processing* node is subscribed to the three *Tag_ID_x* topics. Only when a simultaneous detection is made in the three channels will this node apply its processing. If that's the case the synchronisation is made and only then the DoA is computed.

Chapter 10

Results

Even if many Chapters had preliminary results concerned with each method and technique, for instance the Doa error analysis made in Chapter 7 or the results of each test in the KF calibration, this chapter is responsible of wrapping up all the methods and technologies that were used and/or developed in order to meet the objectives listed in Chapter 1, highlighting the validation of the DoA method and ensure that a target is tracked.

10.1 Simulator

As mentioned in Chapters 5 and 8 two scenarios were created. One where two buoys would form a gate and other where an ASV/boat would follow a target, orbiting it.

Ideally there would not be any errors related with the arrival time in each channel. A simulation of 200 seconds was ran, where the added arrival time errors, depicted in Equation 7.1, are not considered. For the buoys' scenario the target's estimated position has low errors even describing it's trajectory with ease, as seen in Figure 10.1 a), even if the target's velocity is the maximum accepted by the simulation, Figure 10.1 b). This figure depicts a generated simulation where both the target positions and estimated positions are plotted. Figure 10.1 even has the buoys represented with some of the vectors pointing at where the DoA of where the target is. Without any arrival time errors, the results are very pleasing, since besides tracking the target, its position is well computed.

However, errors were added to the arrival times of each channel. Even if the target's trajectory is not described, the errors are reduced for both low target velocity, Figure 10.2 a), and high target velocity Figure 10.2 b). A KF was calibrated for this scenario. Not only did the position estimation errors decrease, but it was also possible to track the target more successfully, as more points were calculated, that is, it did not depend exclusively on the measurements, with positions being estimated by the KF predict phase. A mission of 50 minutes was simulated where the arrival time errors were taken into account, being depicted in Figure 10.3. The filter started as soon as the first measurement was done.

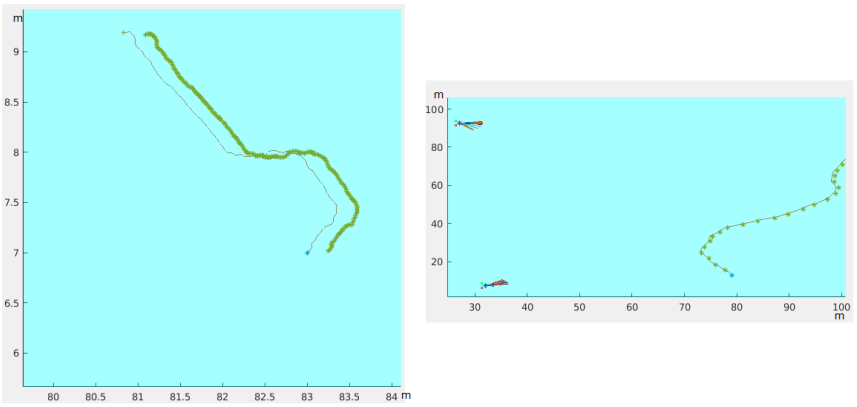


Figure 10.1: Example of a 200 s buoy simulation with no added errors. Left (a) with low target's velocity, Right (b) with with high target's velocity

Instead of having only 3000 measurement dependent positions, a total of 38987 positions were estimated. It is possible to see on the top left of Figure 10.3 the nominal target trajectory in orange and around it the many PE made by the filter. The target was successfully tracked throughout the mission even though its position had changed so much over time, including in depth, as show in Figure's 10.3 bottom. The results can also be more easily perceived through both Table 10.1 and Figure 10.4.

Table 10.1 makes it easier to compare the execution with and without the filter. It is noticeable that the error's mean of the estimated positions decreased considerably , as well as the maximum error. The minimum error was also very low when compared with the simulation that ran without the KF. Figure 10.4 serves the purpose of having a visual representation of the error's magnitude while being compared with the same parameters

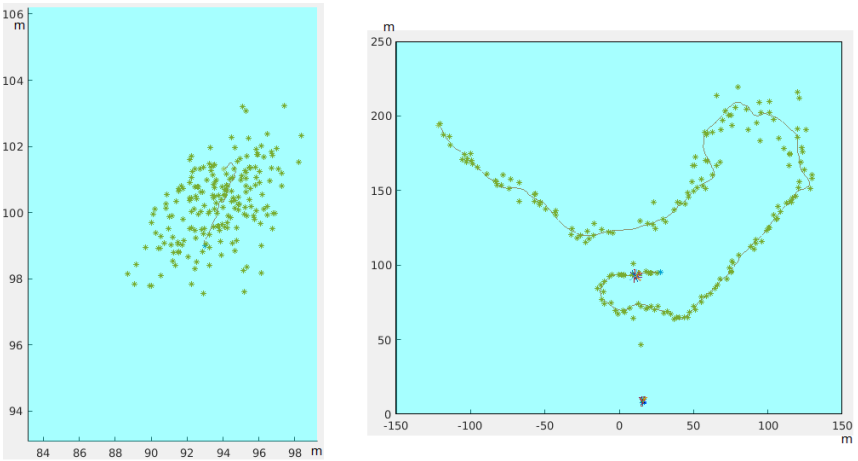


Figure 10.2: Example of a buoy 200 s simulation with added errors. Left (a) with low target's velocity, Right (b) with with high target's velocity

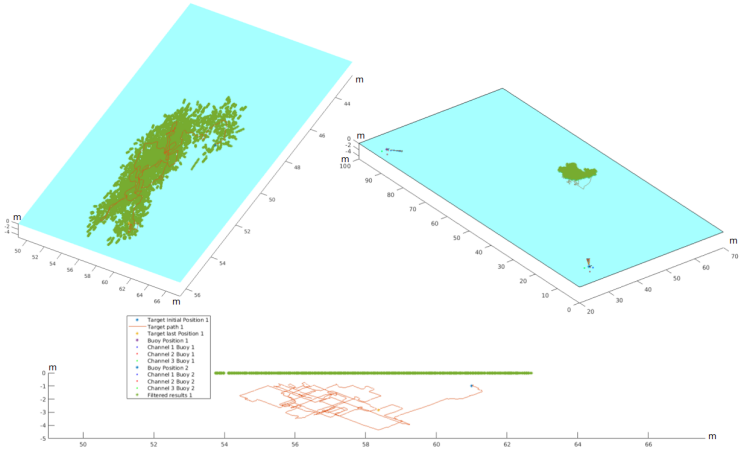


Figure 10.3: Kalman Filter Simulation

as the ones shown in Table 10.1.

Table 10.1: KF on vs KF off error analysis

	Off	On
mean	0.9106 m	0.6438 m
maximum	2.7238 m	2.1495 m
minimum	0.0321 m	0.0018 m

In the boat/ASV scenario for the boat's trajectory to guarantee two different DoA values it needs to be in a different position in relation to where the first DoA was acquired. After some DoA measurements while moving with a random trajectory to avoid false positives, the boat starts to track a previously selected target while describing an orbit trajectory. However, if the target's velocity increases so will the estimated position error, since for the same time interval the target's position is considerably different

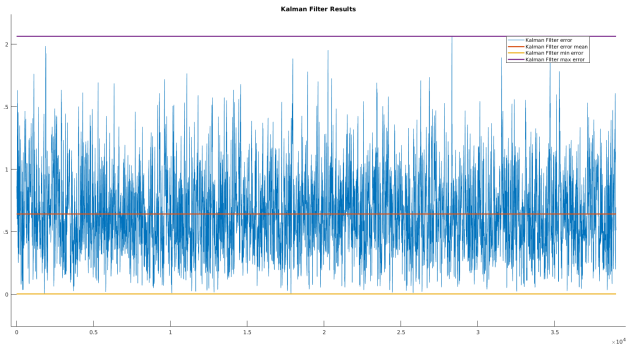


Figure 10.4: Kalman Filter Error representation

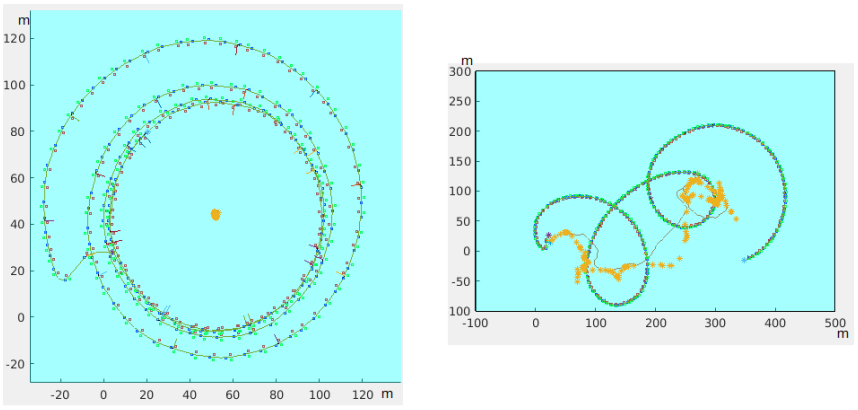


Figure 10.5: Example of a boat 200 s simulation with no added errors. Left (a) with low target's velocity, Right (b) with with high target's velocity

when compared with a slower target, that's why the target's trajectory isn't described as smoothly when compared with the buoys scenario, where the intersection is made at the same instant. Instead of having a perfect orbit around the target with a normal between the boat's direction and the DoA, 3 ° are subtracted to the boat's bearing, increasing the proximity to the target, as mentioned in Chapter 8.

When there aren't any errors related with the arrival time in each channel, the errors of the estimated target's position are low especially when the target's velocity is at its minimum, Figure 10.5 a). Even if it is at its maximum velocity the results are positive, with low errors and, above all, the target is tracked Fig 10.5 b). Even if the estimated

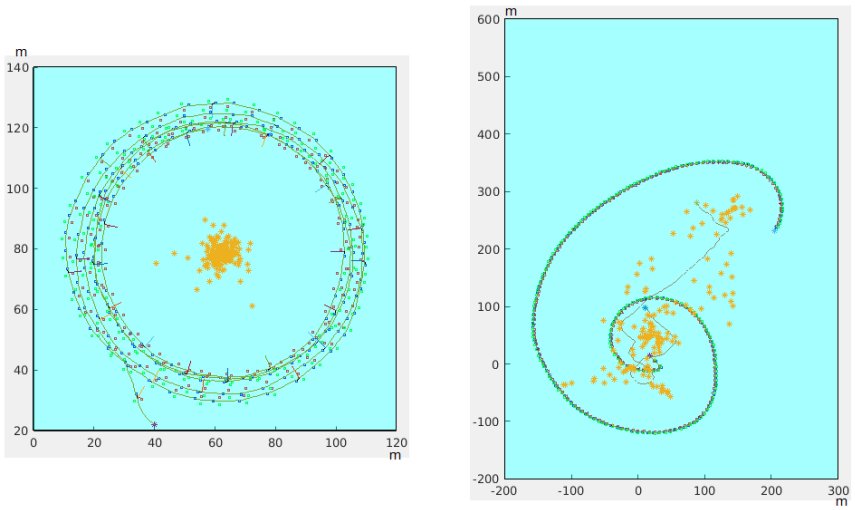


Figure 10.6: Example of a boat 200 s simulation with added errors. Left (a) with low target's velocity, Right (b) with with high target's velocity

position errors increase when the errors are taken into consideration, for both low, Figure 10.6 a), and high velocities, Figure 10.6 b), the target is also tracked successfully.

10.2 Laboratory tests

The hardware and software were validated during the development, as exposed throughout the dissertation, however never in a solution perspective that combined all the necessary parts for its operation. Because of this the following tests were performed in the laboratory's tank.

Figure 10.7 illustrates the setup used in the tests, a set of three receiving systems, such as that described in Chapter 9. Each of the receiving systems was connected to the D-LINK Go ethernet switch, which in turn was connected to the PC that ran the ROS nodes, described in Chapter 9, acquiring the necessary data for processing the signal received by each channel. Each receiver system was powered via Universal Serial Bus (USB) and all of them were connected to the same trigger system. An important step to validate was to confirm that triggers were being detected by each of the receiving systems and this could only be seen from the PC side, since only then was the acquired ADC data stored. Figure 10.8 is an example of data acquired with the reception system before any processing, where among several emissions, there are samples where the value is null and correspond to the instant where the trigger was given, as was explained in Chapter 9, and they are surrounded with a circle for a better understanding. The *ad_receiver* and *Message_Listener* nodes, depicted in Figure 9.25 are thus validated, since data was acquired successfully, sent via Ethernet and with the synchronisation samples amongst the data, as was intended.

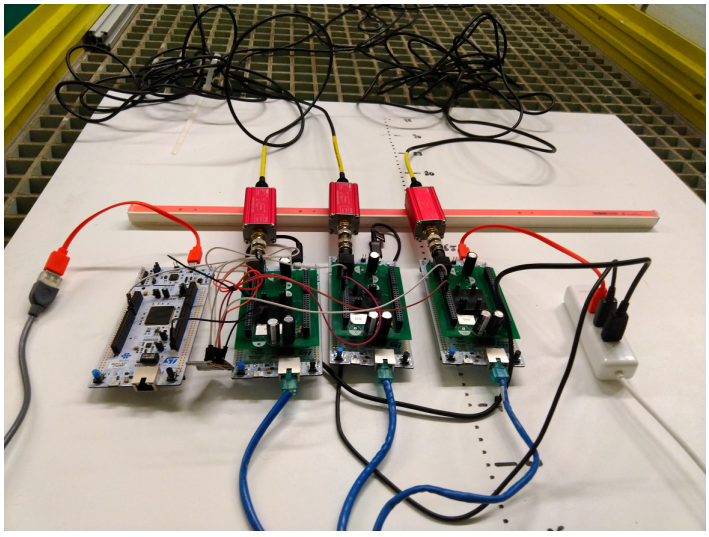


Figure 10.7: Receiver system setup

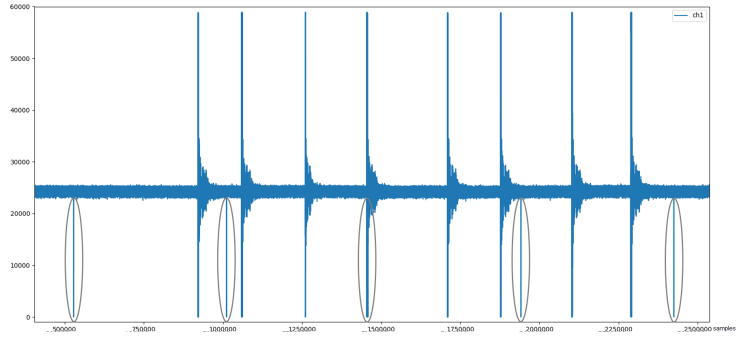


Figure 10.8: uC's acquired data with the synchronisation triggers

In order to the *Bearing_Processer* node to run, all three receivers needed to have a simultaneous Identification of a tag's signal. After that is assured, only then will the DoA be computed. Figure 10.9 demonstrates the results of the ROS nodes executed prior to the DoA computation phase. It is possible to observe the published results of *TagFinder* node for each of the channels, ensuring, besides the simultaneous reception of data the identification of the target's signal. The emitter was also used in these tests and it simulated the sending of the signal from tag 1308 every second, thus justifying why the identifications were so close to one another, proving that identifications can be made in real-time and almost overlapping one another.

To validate if the DoA algorithm was feasible to determine the direction of an acoustic signal and, consequently, to apply the inherent results to estimate the position of targets, tracking them, more specific tests were done.

For this to be possible, it was necessary to place the hydrophones in order to ensure the necessary requirements for the DoA algorithm to function, that is, to have the points that make up the baseline to generate equal angular distances and guarantee that they are equidistant from one another. In addition, it was ensured that each hydrophone was

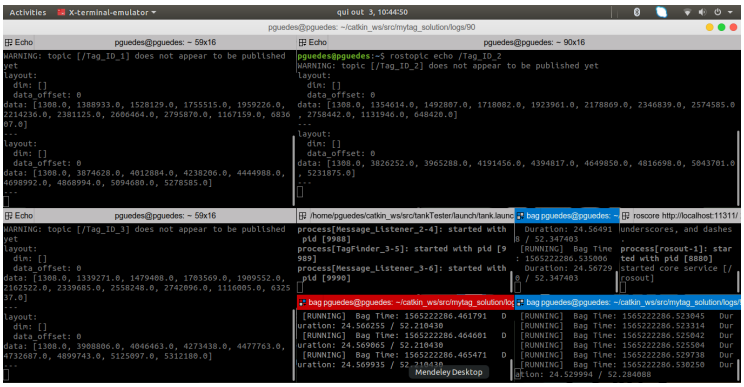


Figure 10.9: Real-time ROS running with the 3 channels

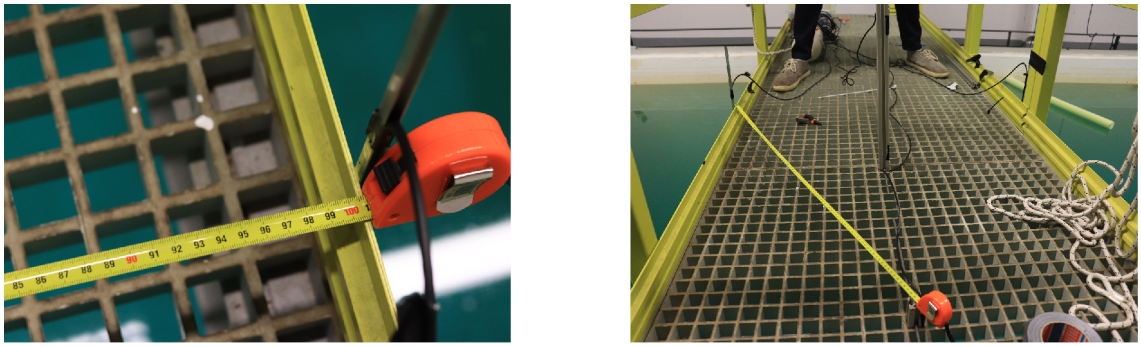


Figure 10.10: Baseline disposition in laboratory's tank

submerged to the same depth to minimise arrival time errors. The same was done to the target, even though it was proven by testing Algorithm 3 in Chapter 7 that it would make no difference how deep the target was in relation to the baseline. Figure 10.10 illustrates what was done, having displayed the baseline with the equilateral triangle's side with 1 m and, consequently, according to Equation 8.25, its centre was equidistant to all vertices and was approximately 0.58 m. In Appendix E more pictures of the tests and baseline measurements are available.

The first test to be done was to look for the error related with delays on the receiving systems. The target was placed in the calculated baseline's centre, as displayed in Figure 10.10. Being at an equidistant point from the receivers, in theory the arrival times would be exactly the same between channels. However, this is not the case due to errors inherent to the environment, as mentioned in Chapter 4, especially multi-path, since the test was done on the tank, many signal reflections occurred on its walls. This would accentuate the

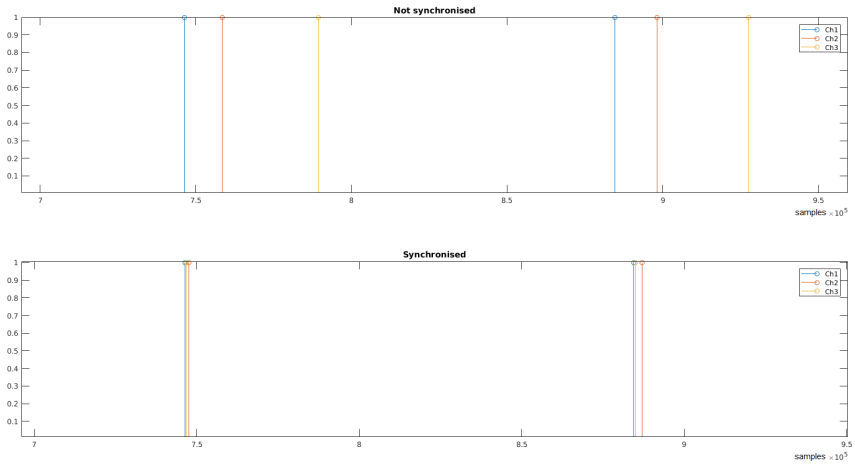


Figure 10.11: Example of a Signal Identification and the differences of not applying vs applying Algorithm 5

number of false positives and affect the signal's correlation, since the correlation's highest peaks can happen when constructive interference occurs, detecting the arrival time with a delay.

Table 10.2: Standard deviation for without and with the application of the synchronisation Algorithm 5

	σ
Not Synchronised	22138 samples
Synchronised	594 samples

Forty simultaneous receptions that were detected by the three reception channels were used to test the delays between them. For this, the standard deviation between channels was computed for each simultaneous identification. The final result was the average of the computed standard deviations. The results of the equidistant distance test were also analysed together with the values obtained without passing through the synchronism algorithm, Algorithm 5. Figure 10.11 is illustrative of the noticeable difference between before applying the algorithm, on top of Figure 10.11, and after its application, Figure's 10.11 bottom.

Applying standard deviation to the three channels in this particular identification gives rise to the results in Table 10.2. With Figure 10.11 and the results of Table 10.2 it is noticeable the difference that the application of the Synchronisation Algorithm 5 makes in the system results, in this case a difference of 21544 samples. Table 10.3 contains the average of all the standard deviations applied to each of the 40 simultaneous measurements that occurred on the three channels after applying the sync algorithm.

In the second test, the target was placed directly in front of the baseline, that is, of channel 1. This test replicates the assumption made in Chapter 7 and illustrated to the right of Figure 7.1, where the sum vector of each resulting vector in the baseline would guarantee a maximum value to the channel on which the target is aligned, since in theory the arrival time at channel 2 and channel 3 would be exactly the same. The results are illustrated in Figure 10.12, where two outliers are notorious. Of the ten simultaneous measurements obtained, the resulting average was approximately 295 ° instead of 360 ° .

Table 10.3: Equidistant test

	σ
max	1591 samples
mean	921 samples
min	85 samples

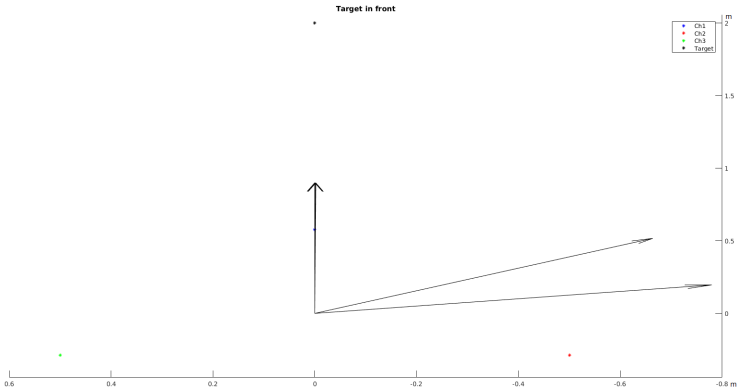


Figure 10.12: Target in front of the baseline

This can be justified given that the target is very close to the baseline where, according to the Algorithm error analysis done in Chapter 7, the errors are higher. However, most of the DoA measurements were approximately 360 °.

Finally a test was made where the target was far from the baseline. The position of the target in relation to the centre of the baseline was given by $[x_{target} = 6.54, y_{target} = 1.59]$ in m. Given these values, the angle generated between the centre of the baseline and the target, according to Equation 10.1, is 13.66 °.

$$tan^{-}1 = \frac{y_{target}}{x_{target}} \tag{10.1}$$

The results are illustrated in Figure 10.13. Out of the fifteen simultaneous measurements that were done, the resulting mean was approximately 15 °.

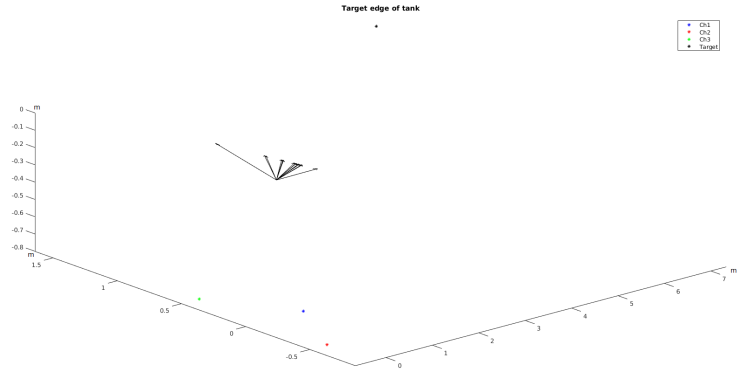


Figure 10.13: Target at the tank's farthest position

THIS PAGE
INTENTIONALLY
LEFT BLANK

Chapter 11

Conclusions

The main objective to be accomplished was to create an underwater acoustic target tracker and this could only be possible if several sub-objectives were met.

According to Chapter 5, it was necessary to develop new hardware and, consequently, new software, highlighting the link between uC, signal conditioning board and the computer. Both a completely built-in emission and reception systems have been developed for better results than those obtained in [3], achieving near to real-time measurements, since all the PC-side software was made with ROS nodes. Both systems were later individually validated, Chapter 9, as well as an integral part of the entire solution, Chapter 10.

In order to track the V7 tags or the developed emitter system, tracking algorithms needed to be developed. For this purpose a DoA Algorithm was created, Algorithm 3, designed to be robust and easy to implement, both in computational terms and in relation to its use in various mission scenarios. This algorithm relied on the use of an equidistant baseline which guaranteed equal angular distances, in the case of this project an equilateral triangle was used to minimise the number of hydrophones. This Algorithm was individually tested through an error analysis performed and referred to in Chapter 7. It has not only been validated in this way, but also through the use of a simulator developed from scratch, Chapter 8. With this simulator it was possible to study two methods designed for the target's position estimation and, consequently, to allow to track them, those being the ASV/boat and the buoys methods. The influence that the KF would have on a simulation environment that contained close-to-reality DoA measurement errors was also addressed. Positive results were obtained in Chapter 10, after the KF calibration performed in Chapter 8.

To be tested in a real scenario, the results of the receiving systems would have to be subjected to the sync Algorithm 5 in order to minimise errors and extract the best possible results from the DoA Algorithm. After all these steps the system as a whole was tested in the laboratory tank and satisfactory results were obtained for the output of the developed DoA method.

That said, the objectives that were proposed in the introduction of the dissertation were successfully met. Knowing this, it is hoped that this document will promote the study in the area of Ocean Robotics and that it has developed the state of the art with regard to acoustic target tracking technologies in the underwater environment.

For eventual future work to improve and reach new levels with the developed solution, the next list follows:

- Improved peak detection and identification method.
- More intensive tank and real-world testing.
- Trigger made with GPS and converting the estimated. positions (relative frame) to the GPS referential (geodetic).
- Implementation of a final watertight hardware solution.
- Implementation of ASV control through way-points.
- Real-time communication between the ASV and a base station and for the buoys scenario.
- Validation of the target tracking in real operation scenario.

Appendix A

Receiver System Schematic

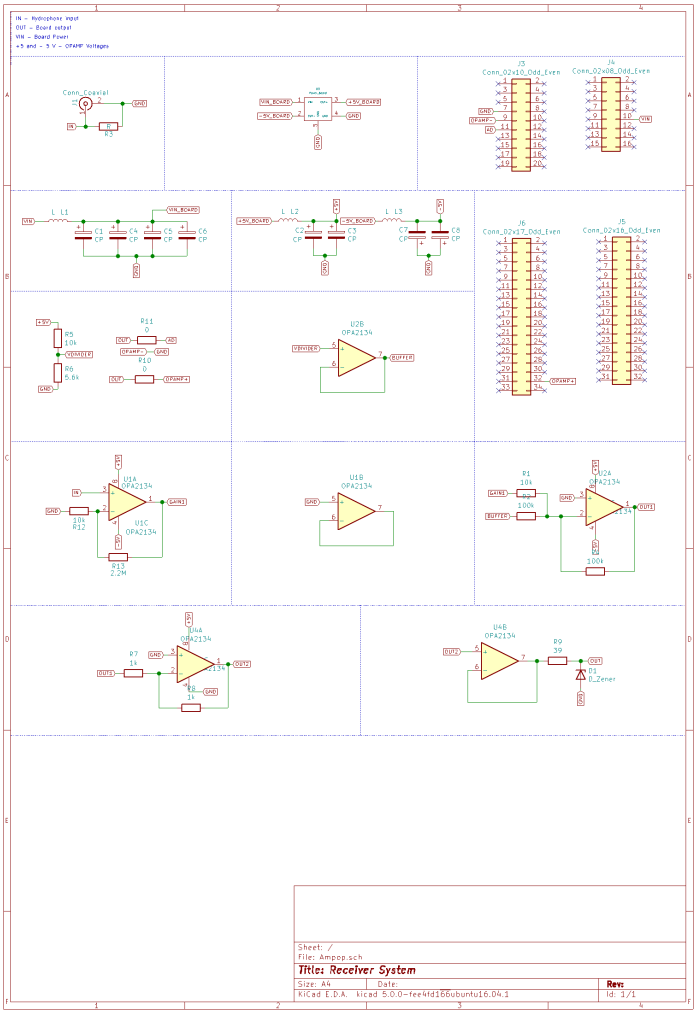


Figure A.1: Receiver system schematic

THIS PAGE
INTENTIONALLY
LEFT BLANK

Appendix B

Power board Schematic

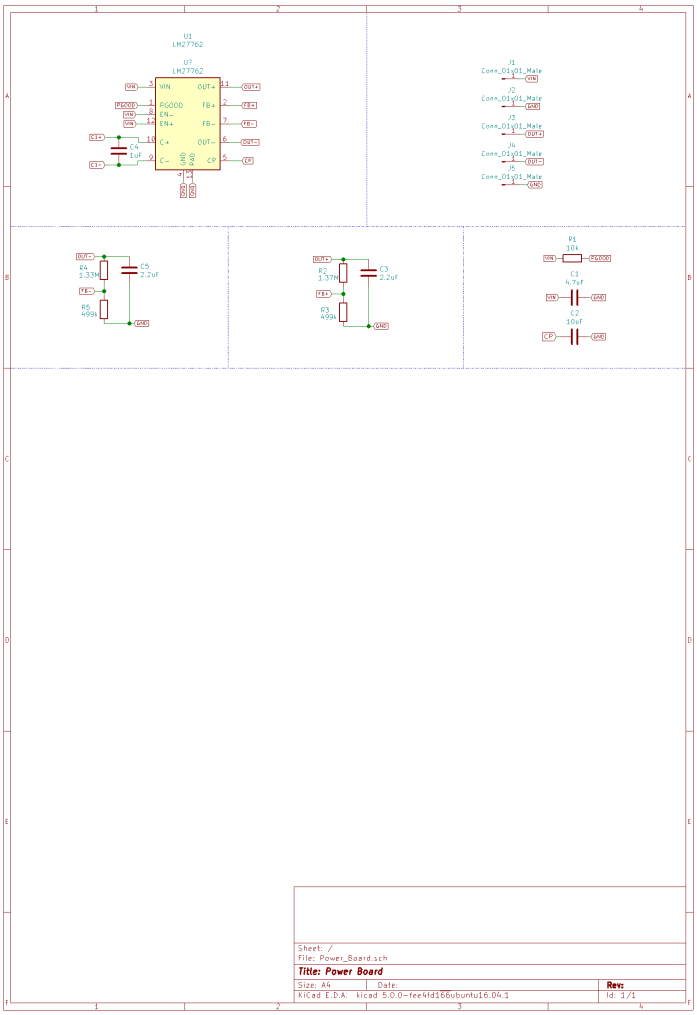


Figure B.1: Power board schematic

THIS PAGE
INTENTIONALLY
LEFT BLANK

Appendix C

Emission system Schematic

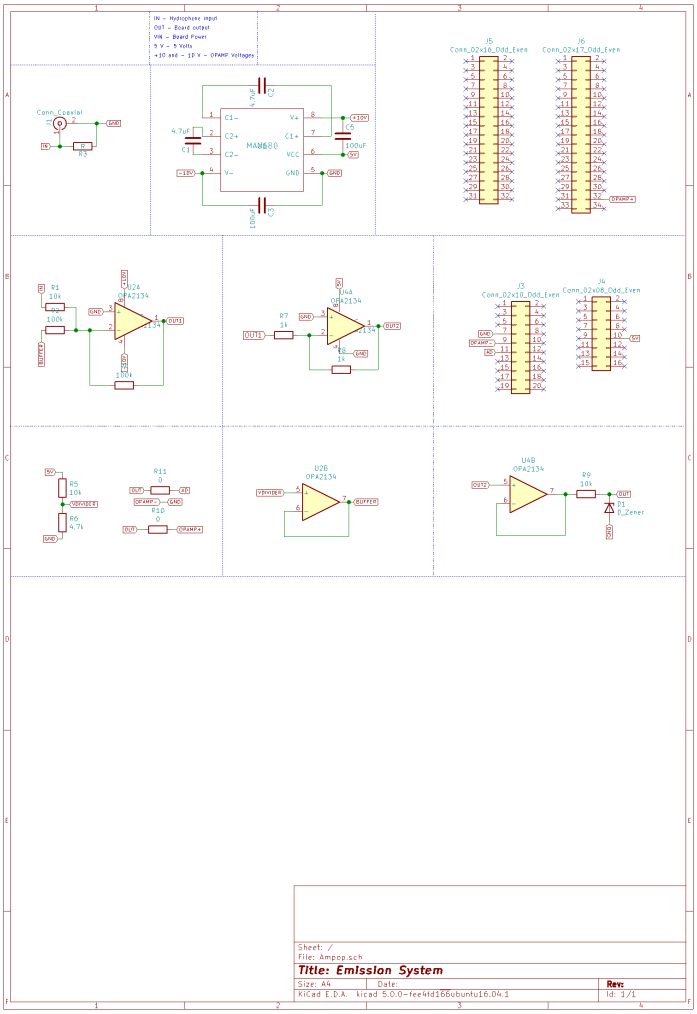


Figure C.1: Emission system schematic

THIS PAGE
INTENTIONALLY
LEFT BLANK

Appendix D

Chi-Square Distribution Table

Chi-square Distribution Table

d.f.	.995	.99	.975	.95	.9	.1	.05	.025	.01
1	0.00	0.00	0.00	0.00	0.02	2.71	3.84	5.02	6.63
2	0.01	0.02	0.05	0.10	0.21	4.61	5.99	7.38	9.21
3	0.07	0.11	0.22	0.35	0.58	6.25	7.81	9.35	11.34
4	0.21	0.30	0.48	0.71	1.06	7.78	9.49	11.14	13.28
5	0.41	0.55	0.83	1.15	1.61	9.24	11.07	12.83	15.09
6	0.68	0.87	1.24	1.64	2.20	10.64	12.59	14.45	16.81
7	0.99	1.24	1.69	2.17	2.83	12.02	14.07	16.01	18.48
8	1.34	1.65	2.18	2.73	3.49	13.36	15.51	17.53	20.09
9	1.73	2.09	2.70	3.33	4.17	14.68	16.92	19.02	21.67
10	2.16	2.56	3.25	3.94	4.87	15.99	18.31	20.48	23.21
11	2.60	3.05	3.82	4.57	5.58	17.28	19.68	21.92	24.72
12	3.07	3.57	4.40	5.23	6.30	18.55	21.03	23.34	26.22
13	3.57	4.11	5.01	5.89	7.04	19.81	22.36	24.74	27.69
14	4.07	4.66	5.63	6.57	7.79	21.06	23.68	26.12	29.14
15	4.60	5.23	6.26	7.26	8.55	22.31	25.00	27.49	30.58
16	5.14	5.81	6.91	7.96	9.31	23.54	26.30	28.85	32.00
17	5.70	6.41	7.56	8.67	10.09	24.77	27.59	30.19	33.41
18	6.26	7.01	8.23	9.39	10.86	25.99	28.87	31.53	34.81
19	6.84	7.63	8.91	10.12	11.65	27.20	30.14	32.85	36.19
20	7.43	8.26	9.59	10.85	12.44	28.41	31.41	34.17	37.57
22	8.64	9.54	10.98	12.34	14.04	30.81	33.92	36.78	40.29
24	9.89	10.86	12.40	13.85	15.66	33.20	36.42	39.36	42.98
26	11.16	12.20	13.84	15.38	17.29	35.56	38.89	41.92	45.64
28	12.46	13.56	15.31	16.93	18.94	37.92	41.34	44.46	48.28
30	13.79	14.95	16.79	18.49	20.60	40.26	43.77	46.98	50.89
32	15.13	16.36	18.29	20.07	22.27	42.58	46.19	49.48	53.49
34	16.50	17.79	19.81	21.66	23.95	44.90	48.60	51.97	56.06
38	19.29	20.69	22.88	24.88	27.34	49.51	53.38	56.90	61.16
42	22.14	23.65	26.00	28.14	30.77	54.09	58.12	61.78	66.21
46	25.04	26.66	29.16	31.44	34.22	58.64	62.83	66.62	71.20
50	27.99	29.71	32.36	34.76	37.69	63.17	67.50	71.42	76.15
55	31.73	33.57	36.40	38.96	42.06	68.80	73.31	77.38	82.29
60	35.53	37.48	40.48	43.19	46.46	74.40	79.08	83.30	88.38
65	39.38	41.44	44.60	47.45	50.88	79.97	84.82	89.18	94.42
70	43.28	45.44	48.76	51.74	55.33	85.53	90.53	95.02	100.43
75	47.21	49.48	52.94	56.05	59.79	91.06	96.22	100.84	106.39
80	51.17	53.54	57.15	60.39	64.28	96.58	101.88	106.63	112.33
85	55.17	57.63	61.39	64.75	68.78	102.08	107.52	112.39	118.24
90	59.20	61.75	65.65	69.13	73.29	107.57	113.15	118.14	124.12
95	63.25	65.90	69.92	73.52	77.82	113.04	118.75	123.86	129.97
100	67.33	70.06	74.22	77.93	82.36	118.50	124.34	129.56	135.81

Figure D.1: Chi-Square Distribution Table

THIS PAGE
INTENTIONALLY
LEFT BLANK

Appendix E

Laboratory test

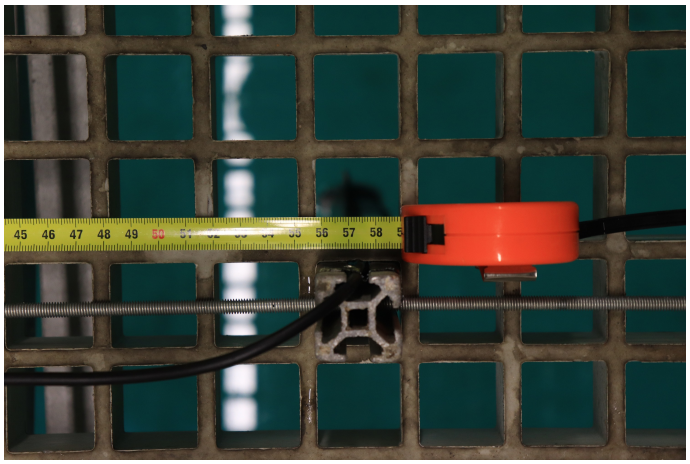


Figure E.1: Distance to the baseline's centre

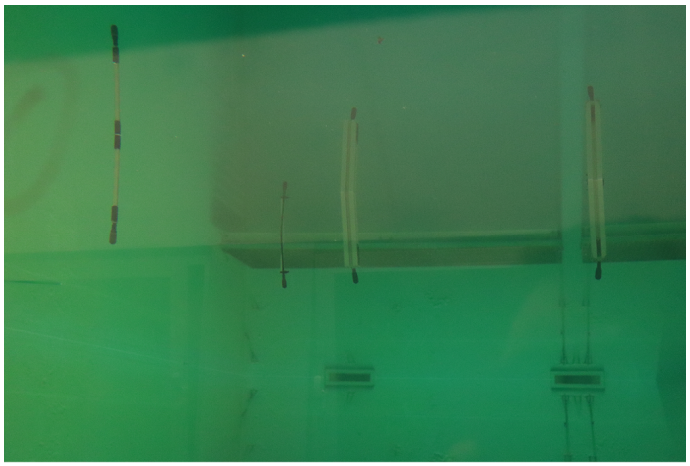


Figure E.2: Submersed hydrophones

THIS PAGE
INTENTIONALLY
LEFT BLANK

References

[1] J. Yuh, G. Marani, and D. R. Blidberg, "Applications of marine robotic vehicles", *Intelligent Service Robotics*, vol. 4, no. 4, pp. 221–231, 2011, ISSN: 18612776. DOI: 10.1007/s11370-011-0096-5.

[2] J. Almeida, A. Ferreira, B. Matias, A. Dias, A. Martins, F. Silva, J. Oliveira, P. Sousa, M. Moreira, T. Miranda, C. Almeida, and E. Silva, "Air and underwater survey of water enclosed spaces for VAMOS! Project", *OCEANS 2016 MTS/IEEE Monterey, OCE 2016*, pp. 5–9, 2016. DOI: 10.1109/OCEANS.2016.7761282.

[3] N. Viana, P. Guedes, D. Machado, D. Pedrosa, and A. Dias, "Underwater Acoustic Signal Detection and Identification Study for Acoustic Tracking Applications", *OCEANS 2018 MTS/IEEE Charleston*, pp. 1–7, 2018.

[4] Vemco, "V7 Coded Transmitters Implantable transmitter for salmon smolt and juvenile species", no. 902, pp. 1–2, 2014.

[5] A. Alcocer, P. Oliveira, and A. M. Pascoal, "Underwater acoustic positioning systems based on buoys with GPS", *In Proc. of the 8th European Conference on Underwater Acoustics ECUA*, no. January, 2006.

[6] P. H. Milne, *Underwater acoustic positioning systems*. Gulf Pub. Co, 1983, p. 284, ISBN: 0872010120.

[7] C. Gernot, "GPS Signal Disturbances by Water in Various States", *... of the 20th International Technical Meeting ...*, no. September 2007, pp. 25–28, 2001. [Online]. Available: http://plan.geomatics.ucalgary.ca/papers/CGernot%7B%5C_%7DGNSS07.pdf.

[8] C. Klungmontri and I. Nilkhamhang, "Acoustic underwater positioning system using fast fourier transform and trilateration algorithm", *ECTI-CON 2017 - 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp. 521–524, 2017. DOI: 10.1109/ECTICon.2017.8096289.

[9] Y. Han, C. Zheng, and D. Sun, "Accurate Underwater Localization Using LBL Positioning System", *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–4, 2015. DOI: 10.23919/OCEANS.2015.7401893.

[10] E. R. Uk, "LBL Underwater Positioning 23/01/2008", pp. 1–3, 2008.

[11] M. J. Crocker, *Handbook of acoustics*. Wiley, 1998, p. 1461, ISBN: 047125293X.

[12] A. Matos, N. Cruz, and F. L. Pereira, "Development and Implementation of a Low-Cost LBL Navigation System for an AUV", pp. 774–779,

[13] J. Cao, C. Zheng, D. Sun, and D. Zhang, "Travel time processing for LBL positioning system", *2016 IEEE/OES China Ocean Acoustics (COA)*, no. 1, pp. 1–5, 2016. DOI: 10.1109/COA.2016.7535752.

[14] K. Vickery, "Acoustic positioning systems. A practical overview of current systems", pp. 5–17, 2002. DOI: 10.1109/auv.1998.744434.

[15] D. Thomson, "Acoustic Positioning Systems a presentation by International Product Manager -Acoustics For The Hydrographic Society in Scotland Hydrofest 2005", 2005. [Online]. Available: http://www.ths.org.uk/documents/ths.org.uk/downloads/2005-04-05-hydrofest-3%7B%5C_%7Dacoustics.pdf.

[16] R. D. Christ and R. L. Wernli, *The ROV manual: a user's guide to remotely operated vehicles*. Butterworth-Heinemann, 2013, p. 308, ISBN: 0080982913.

[17] B. Bell, B. Howe, J. Mercer, and R. Spindel, *Nonlinear Kalman filtering of long-baseline, short-baseline, GPS, and depth measurements*, 2002. DOI: 10.1109/acssc.1991.186428.

[18] S. Smith and D. Kronen, "Experimental results of an inexpensive short baseline acoustic positioning system for AUV navigation", pp. 714–720, 2002. DOI: 10.1109/oceans.1997.634454.

[19] Y. Watanabe, H. Ochi, and T. Shimura, "A study of inverse SSBL acoustic positioning with data transmission for multiple AUV navigation", *Program Book - OCEANS 2012 MTS/IEEE Yeosu: The Living Ocean and Coast - Diversity of Resources and Sustainable Activities*, pp. 1–6, 2012. DOI: 10.1109/OCEANS-Yeosu.2012.6263632.

[20] R. Costanzi, N. Monnini, A. Ridolfi, B. Allotta, and A. Caiti, "On field experience on underwater acoustic localization through USBL modems", *OCEANS 2017 - Aberdeen*, vol. 2017-October, pp. 1–5, 2017. DOI: 10.1109/OCEANSE.2017.8084996.

[21] M. Li Zhou and A. Science, "A Precise Underwater Acoustic Positioning Method Based on Phase Measurement Supervisory Committee A Precise Underwater Acoustic Positioning Method Based on Phase Measurement", 2010.

[22] M. Arkhipov, "An Approach to Using Basic Three-Element Arrays in Tetrahedral-Based USBL Systems", *2013 OCEANS - San Diego*, pp. 1–8, 2013. DOI: 10.23919/OCEANS.2013.6741026.

[23] M. Yu and J. Hui, "The calibration of the USBL transducer array for Long-range precision underwater positioning", *International Conference on Signal Processing Proceedings, ICSP*, pp. 2357–2360, 2010. DOI: 10.1109/ICOSP.2010.5657193.

[24] A. A. P. O. A. Pascoal, "Study and Implementation of an EKF GIB-BASED underwater positioning system",

[25] H. Thomas, "GIB buoys: an interface between space and depths of the oceans", *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles (Cat. No.98CH36290)*, pp. 181–184, 1998, ISSN: 03783820. DOI: 10.1109/AUV.1998.744453. [Online]. Available: http://ieeexplore.ieee.org/document/744453/.

[26] D. Webber, "VEMCO Positioning System (VPS) A low cost , non-real-time underwater acoustic fine-scale positioning system", pp. 2–3, 2009.

[27] R. Roy, J. Beguin, C. Argillier, L. Tissot, F. Smith, S. Smedbol, and E. De-Oliveira, "Testing the VEMCO Positioning System: Spatial distribution of the probability of location and the positioning error in a reservoir", *Animal Biotelemetry*, vol. 2, no. 1, 2014, ISSN: 20503385. DOI: 10.1186/2050-3385-2-1.

[28] V. Kunin, M. Turqueti, J. Saniie, and E. Oruklu, "Direction of Arrival Estimation and Localization Using Acoustic Sensor Arrays", *Journal of Sensor Technology*, vol. 01, no. 03, pp. 71–80, 2011, ISSN: 2161-122X. DOI: 10.4236/jst.2011.13010.

[29] V. KUNIN, "Sound and ultrasound source direction of arrival estimation and localization", PhD thesis, Graduate College of the Illinois Institute of Technology, 2010, pp. 76–99.

[30] A. K. Tellakula, "Acoustic Source Localization Using Time Delay Estimation Dedicated to ...", PhD thesis, Indian Institute of Science, 2007.

[31] J. Stefański, "Hyperbolic Position Location Estimation in the Multipath Propagation Environment", pp. 232–239, 2009. DOI: 10.1007/978-3-642-03841-9_21.

[32] Y. Z. Chan, "Tdoa-sdoa estimation with moving source and receivers", pp. 153–156, 2003.

[33] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location - Signal Processing, IEEE Transactions on", vol. 42, no. 8, pp. 1905–1915, 1994.

[34] W. H. FOY, "Position-Location Solutions by Taylor-Series Estimation", *Aerospace*, no. 2, pp. 187–194, 1976.

[35] A. S. Yaro, M. J. Musa, S. Sani, and A. Abdulaziz, "3D Position Estimation Performance Evaluation of a Hybrid Two Reference TOA / TDOA Multilateration System Using Minimum Configuration", vol. 5, no. 4, pp. 96–102, 2016. DOI: 10.5923/j.ijtte.20160504.03.

[36] T. Melodia, H. Kulhandjian, L.-c. Kuo, and E. Demirors, "Advances in Underwater Acoustic", pp. 804–854, 2013, ISSN: 0041624X.

[37] N. Saeed, A. Celik, T. Y. Al-Naffouri, and M.-S. Alouini, "Underwater Optical Wireless Communications, Networking, and Localization: A Survey", *Ad Hoc Networks*, p. 101 935, 2019, ISSN: 15708705. DOI: 10.1016/j.adhoc.2019.101935. arXiv: arXiv:1803.02442v1.

[38] Luís Carlos da Cunha Vides Gonçalves, "Underwater Acoustic Communication System: Performance evaluation of digital modulation techniques", 2012.

[39] N. Farr, N. Farr, A. Bowen, J. Ware, and C. Pontbriand, "An integrated , underwater optical / acoustic communications system An integrated , underwater optical / acoustic communications system", *IEEE Xplore*, no. June, 2010, ISSN: 17683254.

[40] X. Che, I. Wells, G. Dickers, P. Kear, and X. Gong, "Re-evaluation of RF electromagnetic communication in underwater sensor networks", *IEEE Communications Magazine*, vol. 48, no. 12, pp. 143–151, 2010, ISSN: 01636804. DOI: 10.1109/MCOM.2010.5673085.

[41] M. Stojanovic, "Underwater Acoustic Communication", *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–12, 2015. DOI: 10.1002/047134608X.W5411.pub2. [Online]. Available: http://doi.wiley.com/10.1002/047134608X.W5411.pub2.

[42] S. Mann and S. Haykin, "The Chirplet Transform: A Generalization of Gabor's Logon Transform", *Vision Interface '91*, pp. 205–212, 1991.

[43] T. Smyth, "Linear Frequency Modulation (FM) Chirp Signal Linear Chirp Signal Vibrato simulation Vibrato cont .", 2013.

[44] *Chirp, https://en.wikipedia.org/wiki/chirp/media/file:linear-chirp.svg.*

[45] *Cylindrical vs. Spherical Spreading Discovery of Sound in the Sea.* [Online]. Available: https://dosits.org/science/advanced-topics/cylindrical-vs-spherical-spreading/.

[46] S. Kah and H. Wong, "Communications", 2005.

[47] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Journal of Fluids Engineering, Transactions of the ASME*, vol. 82, no. 1, pp. 35–45, 1960, ISSN: 1528901X.

[48] I. Reid and H. Term, "Estimation II Discrete-time Kalman filter", *System*, pp. 1–44, 2001.

[49] A. Scientific and B. Measurement, "AS-1 Datasheet",

[50] S. R. Data, L. Impedance, I. Voltage, and S. Temperature, "50 kHz High-Pass Filter", 2016.

[51] STMElectronics, “Nucleoh743zi datasheet”, no. April, 2019.

[52] D. Information, “OPAx192 36-V , Precision , Rail-to-Rail Input Output , Low Offset Voltage , Low Input Bias Current Op Amp with e-trim”, 2015.

[53] M. Integrated, “Max680 datasheet”,

[54] D. Information, “LM27762 Low-Noise Positive and Negative Output Integrated Charge Pump Plus LDO”, 2017.

[55] STMicroelectronics, 2016. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/datasheet/65/cb/75/50/53/d6/48/24/DM00141306.pdf/files/DM00141306.pdf/jcr:content/translations/en.DM00141306.pdf.

[56] W. D. S. Morgan Quigley, Brian Gerkey, *Robots with ROS*. 2015, p. 447, ISBN: 9781449323899.