COPPE
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
UFRJ

UNDERWATER LOCALIZATION USING IMAGING SONARS IN 3D ENVIRONMENTS

Gabriel Alcantara Costa Silva

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Ramon Romankevicius Costa

Rio de Janeiro
Março de 2017

UNDERWATER LOCALIZATION USING IMAGING SONARS IN 3D ENVIRONMENTS

Gabriel Alcantara Costa Silva

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Ramon Romankevicius Costa, D.Sc.

Prof. Alessandro Jacoud Peixoto, D.Sc.

Prof. Antonio Candea Leite, D.Sc.

Prof. Paulo Cesar Pellanda, Dr.ENSAE

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2017

Silva, Gabriel Alcantara Costa

Underwater Localization using Imaging Sonars in 3D environments/Gabriel Alcantara Costa Silva. – Rio de Janeiro: UFRJ/COPPE, 2017.

X, 70 p.: il.; 29, 7cm.

Orientador: Ramon Romankevicius Costa

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2017.

Bibliografia: p. 63 – 70.

1. Localization. 2. Particle Filter. 3. Imaging Sonar. 4. Sonar Model. 5. Range Model. 6. Sonar Simulation. 7. 3D Environment. I. Costa, Ramon Romankevicius. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

Agradeço a minha família, aos meus pais Telma e Eduardo e a minha irmã Joana, que possibilitaram e contribuiram para a minha formação como pessoa e me ensinaram os valores que me acompanham em minha carreira profissional e acadêmica.

Agradeço a minha companheira, noiva e amiga, Fernanda Nunes, que me fez enxergar o mundo de uma forma mais suave e bela. Agradeço principalmente o suporte que me ajudou a sobreviver ao doloroso processo de escrita, me acalmando com a experiência de quem já passou por isso e também gerenciando eficientemente uma reforma completa em dois banheiros!

Agradeço ao meu chefe e amigo Sylvain Joyeux por toda a orientação técnica e emocional que foram cruciais para o desenvolvimento deste trabalho. Mesmo sem totalmente apreciar o "*brazilian humor*" teve a paciência e atenção em me ajudar.

Agradeço a minha equipe de trabalho que aturou minhas variações de humor e também fez essa experiência um pouco mais prazerosa. Em especial a Eduardo Elael que compartilhou comigo o infortúnio sincronismo entre viagens de campo, provas e prazos.

Agradeço aos meus amigos (e padrinhos) por compreenderem a minha ausência e em especial ao Arthur por dar aquela força nos momentos finais com uma revisão bem colocada.

Agradeço à *Energia Sustentável do Brasil* (ESBR) pela oportunidade de trabalhar no projeto ROSA (Contrato Jirau 151/2013 P&D ANEEL 66310002/2013), que possibilitou e motivou a abordagem deste tema.

Por fim, agradeço aos membros da banca, professores Alessandro Jacoud Peixoto, Antonio Candea Leite, Paulo Cesar Pellanda pela disponibilidade e interesse em avaliar esse trabalho e ao meu orientador Ramon Romankevicius Costa pela devida orientação no decorrer dessa jornada.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

LOCALIZAÇÃO SUBMARINA UTILIZANDO UM SONAR TIPO *IMAGING* EM AMBIENTES 3D

Gabriel Alcantara Costa Silva

Março/2017

Orientador: Ramon Romankevicius Costa

Programa: Engenharia Elétrica

Este trabalho propõe um método de localização utilizando um sonar do tipo MSIS (*Mechanically Scanned Imaging Sonar*), o qual se destaca por seu baixo custo e peso. O método implementa um filtro de partículas, um estimador Bayesiano, e introduz um modelo de medição baseado na teoria de simulação de sonares. No conhecimento do autor não há uma abordagem similar na literatura, uma vez que os métodos atuais de simulação de sonar visam a geração de dados sintéticos para o reconhecimento de objetos. Esta é a maior contribuição da tese pois permite a a computação dos valores de intensidade fornecidos pelos sonares do tipo *imaging* e ao mesmo tempo é compatível com os métodos já utilizados, como extração de distância.

Simulações mostram o bom desempenho do método, assim como sua viabilidade para o uso de *imaging sonars* na localização submarina. A nova abordagem tornou possível, sob certas restrições, a extração de informações 3D de um sensor considerado, na literatura, como somente 2D e também em situações em que não há nehnuma referência no mesmo plano horizontal do eixo de escaneamento do transdutor. A localização em ambientes 3D complexos é também uma vantagem proporcionada pelo método proposto.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

UNDERWATER LOCALIZATION USING IMAGING SONARS IN 3D ENVIRONMENTS

Gabriel Alcantara Costa Silva

March/2017

Advisor: Ramon Romankevicius Costa

Department: Electrical Engineering

This work proposes a localization method using a mechanically scanned imaging sonar (MSIS), which stands out by its low cost and weight. The proposed method implements a Particle Filter, a Bayesian Estimator, and introduces a measurement model based on sonar simulation theory. To the best of author's knowledge, there is no similar approach in the literature, as sonar simulation current methods target in syntethic data generation, mostly for object recognition . This stands as the major contribution of the thesis as allows the introduction of the computation of intensity values provided by imaging sonars, while maitaining compability with the already used methods, such as range extraction.

Simulations shows the efficiency of the method as well its viability to the utilization of imaging sonar in underwater localization. The new approach make possible, under certain constraints, the extraction of 3D information from a sensor considered, in the literature, as 2D and also in situations where there is no reference at the same horizontal plane of the sensor transducer scanning axis. The localization in complex 3D environment is also an advantage provided by the proposed method.

Contents

List of Figures

Chapter 1

Introduction

1.1 Contextualization

The development of underwater systems has been grown remarkably in the last years, due to the increase in the complexity and risk of the tasks to be performed. The inspection, installation and maintenance of underwater structures can be performed by a diver. However, these tasks involve life danger, are slow, expensive and limited by the depth a human diver. To fulfill the increasing needs of the oil and gas and hydroeletric industries, as well as deep-sea research, the academic and industrial communities are focusing on the development of unmanned underwater vehicles (UUVs) and systems. In this context, the GSCAR (Grupo de Simulação e Controle em Automação e Robótica)[1] group, including the participation of the Laboratório de Controle (LABCON)[2] and the Laboratório de Controle e Automação and Engenharia de Aplicação e Desenvolvimento (LEAD)[3], both part of Electrical Engineering Department (PEE/COPPE/UFRJ), has given a special attention to submarine robotics. In 2004, the group started the development of a Remotely Operated Vehicle (ROV), called LUMA [6]. The ROV was initially designed to operate in the inspection of hydroelectric intake tunnel inspection, however in 2007 it was modified for performing Census of Marine Life in Admiral Bay, in the Antarctic continent and was sucessful in three expeditions.

In 2013, an underwater system, called ROSA (*Robô para Operações em Stoplogs Alagados*), was developed in partnership with the *Energia Sustentável do Brasil* (ESBR)[4] . The objective of the system is to monitor operations with Stoplogs at the hydroelectric power plant of Jirau, at the Madeira River. Stoplogs are modular metal beams that are placed on top of each other to allow the depletion of a hydroelectric

[1] http://www.coep.ufrj.br/gscar/

[2] http://www.coep.ufrj.br/gscar/labcon.html

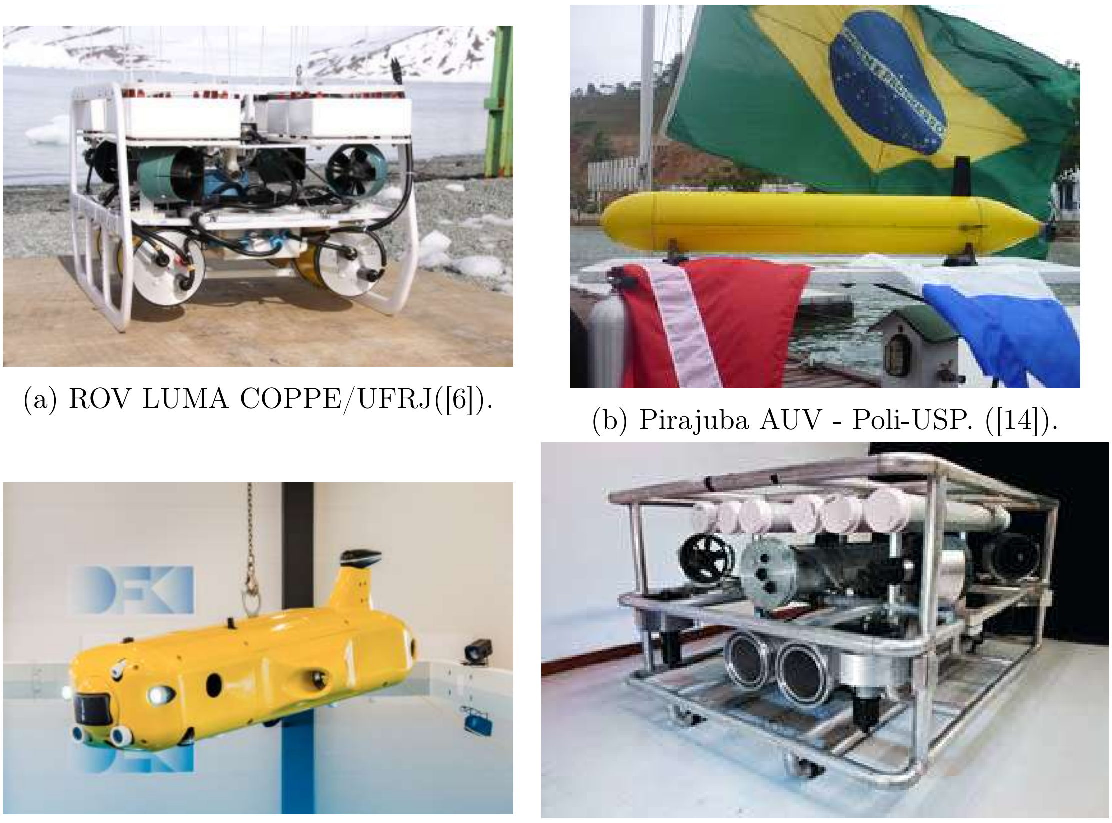
[3] http://www.coep.ufrj.br/gscar/lead/

[4] http://www.energiasustentaveldobrasil.com.br/

turbine. To ensure a tight seal between the first *Stoplog* and the channel bottom, the system uses a sonar to map the underwater environment and detect the presence of debris large enough to interfere with the correct placement of the *Stoplog.* ROSA Robot consists of a system with a ground and an underwater embedded electronics. The latter has four inductive sensors, a pressure sensor, inclination sensors, a pan-tilt unit and the possibility to connect either a profilling sonar, the *Tritech Seaking*, or a Mechanically Scanned Imaging Sonar (MSIS), the *Tritech Micron.* The ROSA project was the motivation for this work as presented interesting technical challenges, such as level variation of the Madeira River and the silt suspension made necessary a localization system for this underwater robot.

Figure 1.1: ROSA project with ESBR and COPPE teams.

Internationally, there is a wide variety of UUVs for different purposes. Besides the already two mentioned vehicles developed by our group at the *Federal University of Rio de Janeiro*(UFRJ) , it is possible to highlight some of the most important and recent work in this field in the national (Brazilian) scenario. The Unmanned Vehicles Laboratory at University of Sao Paulo developed a low cost *Autonomous Underwater Vehicle*(AUV) for investigations on dynamics and navigation of this class of vehicle. The AUV, called Pirajuba, has torpedo-like shape and is approximately 1.75m long [14]. The ROV Laurs, from the *Laboratório de Ultra-som e Robótica Submarina* at USP, was developed to perform deep sea mission as the retrieve of acoustic transponders. Recently, a partnership between the BG Group Brazil and the Brazilian Institute of Robotics in Salvador started the FlatFish project, which aims the development of a subsea-resident AUV to inspect the infrastructure of oil and gas sites on demand. The Figure 1.2 presents a glimpse on the national panorama of unmaned underwater vehicles in Brazil.



(a) ROV LUMA COPPE/UFRJ([6]).

(b) Pirajuba AUV - Poli-USP. ([14]).

(c) FlatFish AUV - BG/BIR([1]).

(d) LAURS *Laboratório de Ultra-som e Robótica Submarina*-USP

Figure 1.2: Some of the UUVs developed in Brazil.

1.2 Objectives

In order to perform its desired tasks and missions, an underwater robot must know its own location relative to the objects to be manipulated. Position estimation, or the so called localization problem consists, therefore, in finding the transformation or correspondence between the robot's local coordinate system and a global reference coordinate system. The problem is not trivial, as the position of the robot usually cannot be sensed directly. And this is specially true in the underwater scenario, where sensors that are taken for granted at the surface (LIDAR, GPS, etc) are simply not available as they use very high frequencies, which have a high attenuation gradient in underwater environments. Therefore, most underwater systems must rely on acoustic based devices. Sonars plays a fundamental role in underwater localization as this devices allow to gather information from the environment without the need of auxiliary infrastructure.

The main objective of this work is to propose a localization method for a generic undewater vehicle using a *Mechanically Scanned Imaging Sonar*. This type of sonar is a common choice among small vehicles such as ROVs and AUVs, due to its low weight and cost. The intrinsec characteristics of an imaging sonar such as the wide vertical opening of its sonar beam well suits this type of sonar to perform obstacle avoidance, however introduces ambiguity in the readings. The current methods present in the literature exploit the mature range finder theory and represent the sonar beam as a unidimensional beam. As the main contribution of this work, sonar simulation theory is embedded in the localization loop and allowing the representation of the shape of an MSIS sonar beam and its interactions with the environment. This detailment in the representation allows the algorithm to estimate the position of underwater systems in complex three dimensional environments and also to extract information outside the main axis of the sensor.

1.3 Outline of the Thesis

- Chapter 2: *Literature Review: Underwater Localization* This chapter presents an overview of the Underwater Localization field and the most recent techniques applied. The principle of operations of sonar devices are also described in this chapter, as well the methods applied to perform localization using different types of sonar devices. An analysis of the sonar beam processing is performed and the limitations present in the current methods is defined.

- Chapter 3: This chapter describes the proposed method to localize an underwater vehicle using a *Particle Filter* and a new measurement model for an imaging sonar, embedding sonar simulation to cast reference beams. The reader can find also the mathematical background of stochastic state estimation at the begining of the chapter.

- Chapter 4: The implementation of the proposed method is discussed in this chapter, as well as its software overview. The results of the method are shown at the end of the chapter.

- Chapter 5: Conclusion and future work are explored in this chapter.

Chapter 2

Literature Review: Underwater Localization

Localization is the process of estimating the pose of a rigid body within a given reference coordinate system. Sensors that provide this pose directly, with the robustness, noise and frequency characteristics required by the robot's actions, are usually not available. This is usually solved by *fusing* the information of multiple sensors into an estimate of the vehicle's pose.

How hard the localization process is can depend on several aspects([63]):

- the knowlegde available about the initial state, i.e robot's initial position;
- how dynamic the environment is;
- if the localization algorithm will have right to control the robot's actions to aid the estimation of the system's pose.

Regarding the knowlegde of the robot's initial state, the localization problem can be classified in three general categories, presented in order of complexity:

- **Position tracking** - the initial pose of the robot is considered to be known, or at least an assumption of the real pose is made. The system must be able only to track the robot pose and handle noise present in the measurements, which is usually considered small and often approximated by an unimodal distribution (e.g., a Gaussian). This scenario is called a **local localization** problem, as the uncertainty is bounded to a region around the robot's true pose.
- **Global localization** - The initial pose of the robot is now unkown and the system must be able to localize itself relying only on the readings of its sensors. This relaxation on the initial pose knowledge makes the problem more complex, because the error cannot be assumed to be bounded and the representation of uncertainties are often not unimodal anymore.

- **Kidnapped robot problem** - This problem has the same assumptions on the initial pose of the robot as the localization problem, but considers that the robot can be moved or teleported to another position during its operations, with the "kidnapping" not being announced to the robot. This means that the system must be able to recover its true pose, regardless of its current belief about its own state. This is useful to evaluate the capability of the system to recover from a system failure.

Regarding the environment's dynamicity, in *static environments*, only the robot is considered to move while the environment and all its features are still. In opposition, in *dynamic environments*, some elements can change position or state, for instance moving people, moving furniture or elements. This aspect influences how the current information about the environment can be used to infer the robot position if compared with the previous known map. In very dynamic environments, the localization problem cannot be separated from a mapping problem.

Regarding whether the robot actions are directly involved in the sensing process, there are two classifications: *passive* and *active*. If the localization algorithm can interfere in the control of the robot in order to gather specific informations about the environment, or to move the robot out a region of high ambiguity in the readings, the localization is defined as *active*. Otherwise, if the localization algorithm can only observe the readings and has no further control on the robot's action, the system is defined as *passive*.

Localization in an underwater environment is particularly challenging due to rapid attenuation of high frequency electromagnetic signals, poor water visibility and the unstructured nature of the environment itself. This makes the implementation of general and broad used strategies applied in ground and aerial systems, as GPS for global localization, ineffective in an underwater scenario. To overcome this limitations, acoustic based sensors are mainly used. As summarized in [49], underwater localization techniques can be organized with respect to the kind of sensor used, the auxiliary infrastructure needed and the information available about the environment. Dead reckoning only uses sensors that give information about the robot's own state (speed). To augment dead reckoning, acoustic transponders and modems can be used to determine the robot's position by measuring the *Time of Flight* of the acoustic signals. Finally, one can also gather data from the environment to estimate the position of the system.

2.1 Dead reckoning

When the system relies only on internal sensors and do not aquire any information about the surrounding environment, a robot must use information of its own movement to be able to estimate its actual position.

The fundamental idea of *Dead Reckoning* (DR) is the integration of incremental movement over time, with the prior knowledge of a previous valid position. This kind of position estimation provides a good short term accuracy. But, due to continuous integration of each uncertainty, it introduces an unbounded cumulative error overtime.

Inertial navigation systems (INS) integrate the linear and angular velocities and the accelerations provided by embedded sensors to estimate the motion of the system. There is a wide variety of inertial sensors available in the market with a wide price range. In general, the price of the sensor is related with the quality and precision of measurements and to which depth the sensor is rated for. The most common sensors applied in DR of underwater systems are the following:

- **Doppler Velocity Log (DVL)** - The DVL is a hydroacoustic sensor that measures the relative velocity between the instrument and the sea bottom (Fig. 2.1). The principle of operation of the DVL is to transmit downlooking acoustic pulses and analyse the phase shift caused by the doppler effect. A DVL can provide velocity measurements with a precision of 0.3% or less with a range of 18 to 100 meters [38]. It can have one or more transducers and with at least three the sensor is capable of provide a 3D velocity vector. Another sensor that uses the same principle, but measures the echoes of pulses reflected by suspended particles is the *Acoustic Doppler Current Profile* (ADCP or ADP) . As the name suggests, this device is used to estimate the water current velocity.

Figure 2.1: Examples of different DVL models, each one best suited for a specific depth and range. ([7]).

- **Inertial Measurement Unit (IMU)** - This is an integrated system that incorporates accelerometers, gyroscopes and compass in order to provide the vehicle's orientation and its velocities, as well the gravitional forces. The compass gives the heading, or the yaw measurement, of the system based on Earth's magnetic field. This kind of sensor is sensitive to objects with a strong magnetic signature and can, therefore, be biased. Gyroscopes are responsible for measuring the angular rates of the vehicle and accuracy and drift will vary on the type of gyroscope used. *Fiber Optic Gyros* (FOG) and *Ring Laser Gyros* (RLG), based on *Sagnac interference*, provide a drift as low as $0.0001^o/h$, while solid-state *Micro-Electro-Mechanical Sensors* (MEMS) can introduce a drift error of $60^o/h$. However due to its small size, weight and cost, MEMS based IMUs are popular among small ROVs and AUVs. The accelerometers measure the forces that actuate on the vehicle, including the gravitational force, and the bias in MEMS system can vary from 0.001 to 1 micro-g [13].

- **Pressure sensors** - this kind of sensor directly measure the vehicle depth in an underwater environment (Fig 2.2). The height of the water column above the sensor can be calculated with high accuracy, due to the much steeper pressure gradient in water, if compared with air. There are two common types of this sensor: Strain gauge sensors, which employ metal alloys or silicon crystals. The resistance of this sensors changes linearly with the pressure applied and have an overall accuracy up to about 0.1% of full-scale and resolutions up to 0.001%. The other most common type is a quartz crystal pressure sensor, which can achieve 0.01% and 0.0001% of overall accuracy and resolution at full-scale, respectively [38]. It is noteworthy that the sensor must be calibrated and the measurement takes into account only the height of the water column above the device, if there are changes in the absolute depth of water in the environment as an effect of tides for example, the true position along the $z - axis$ will be biased.

Figure 2.2: Pressure sensor used in the ROSA and Luma projects ([66]).

As a result of the cumulative errors introduced by continuous integration of measurements, an INS is usually applied in combination with others sensors that can measure or observe the position (or portion of it) of the system and therefore bound the error. Inertial systems use an internal motion model and, with its high sampling rate capability, provides a motion prediction until the measurement of the complementary and slower sensor arrives. Some devices which can measure, even indirectly, the position or pose of the vehicle will be described in the next sections

2.2 Acoustic transponders and beacons

Localization using acoustic transponders is achieved by measuring the *Time-of-Flight* (TOF) of artificially placed acoustic signal transmitters. Techniques differ basically with respect to the geometry of the transponder placement. This strategy has a significant implementation cost, and is also limited to a certain area, as it needs the placement of beacon(s). The beacons can be organized closed together as in the Ultrashort and Short Baseline configuration (USBL and SBL), or in a more sparse way and covering a larger area as in Long Baseline (LBL) and GPS intelligent buoys.

2.2.1 Ultrashort and Short Baseline

Ultrashort Baseline(USBL) uses an array of transceivers placed close together with a spacement on the order of less than $10cm$ and estimates the relative position calculating range with the *Time fo Flight* of the emitted acoustic signal. While elevation and azimuth angles are calculated based on the phase difference across different transponders of the array. The transceiver is usually attached to a surface ship and the transponder is located at the vehicle being tracked, as illustrated in Figure 2.3. In constrast, *Short Baseline* or SBL explore the full length of the support ship and place the transceivers (at least 3) spaced along its hull. That allows the system to analyse the difference in time of arrival between the multiple acoustic pings to estimate bearing, while range is calculated by TOF. The basic configuration of a SBL system is illustrated in Figure 2.4.

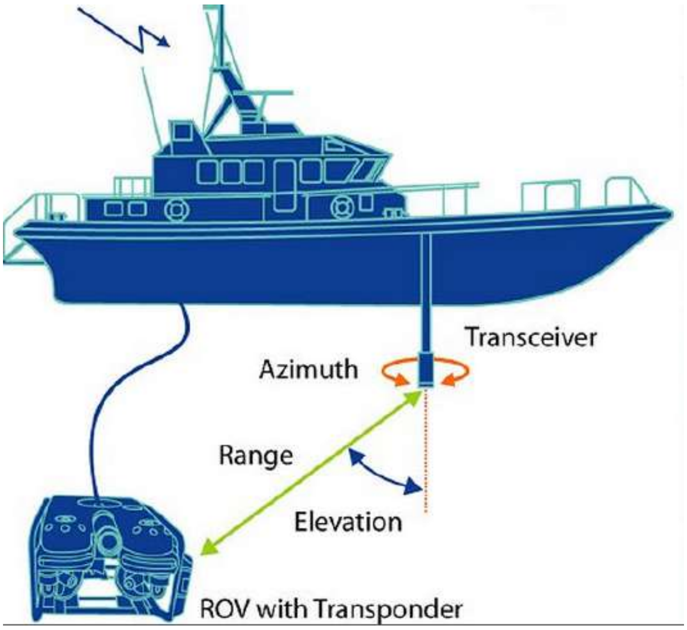


Figure 2.3: Basic configuration of an USBL system. ([64]).

Communication in an USBL system employes only one transceiver, usually attached to the support ship, and the transponder coupled to the underwater vehicle. Therefore, range is limited by the reach of this communication layer. On the other hand, in SBL systems, is the size of the baseline, i.e the size of the support ship, is what dictates how spaced the transmitters are and the positional accuracy is highly affected by this property [49].

In order to georeference the target vehicle, the position of the UBSL transceiver must be known within the reference frame. This means that a calibration step is needed to integrate the localization systems from the support vessel (GPS for example) and the USBL system. *Ridao et al.* proposes, in [56], a localization system for an AUV to perform mapping and inspection of hydroelectric dams. The

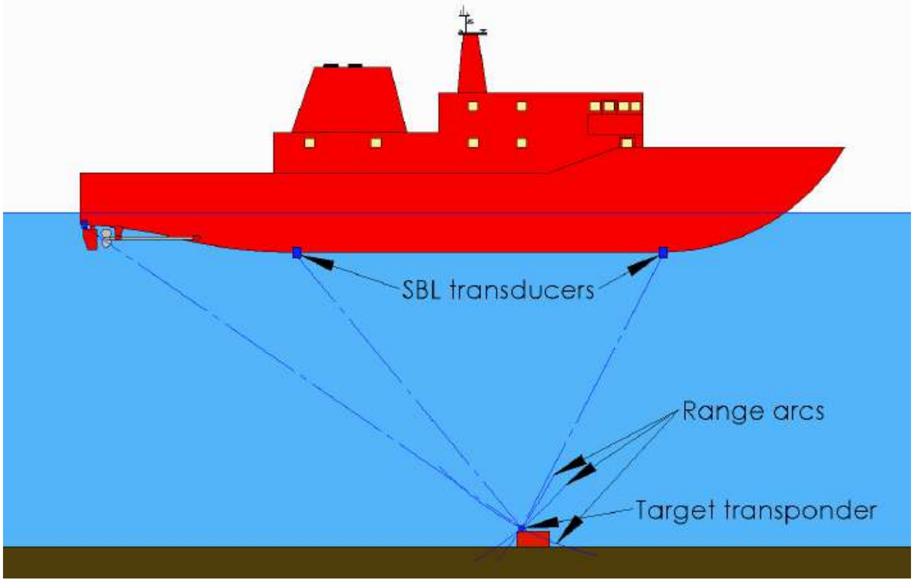


Figure 2.4: Basic configuration of a SBL system. ([47]).

USBL transceiver is coupled to a surface buoy responsible for providing absolute position in world coordinates. The buoy is equipped with a differential GPS and has a dedicated INS, used to correct the movement of the USBL transceiver during the estimation of the AUV position.

The arrangement of the USBL transceivers array and the acoustic transponder is not fixed and different strategies can be applied, depending on the kind of application. *Batista et al.* [3] developed a homing controller to an AUV in which the underwater vehicle had only to localize with respect to one fixed target. Based on that, the USBL transceiver was attached to the AUV itself, and a fixed acoustic transponder was placed at the dock. This has allowed the AUV to calculate directly the distance and the direction to the docking station. A similar approach, with a unique transponder fixed on a ship, was implemented by [69] to make possible the simultaneous positioning of multiple underwater vehicle with respect to one ship.

To overcome the range limitation, *Khan et.al* [37] applied an acoustic modem to augment the USBL system's reach. The authors fused the information of both devices in order to achieve a system with larger range and more robust to noise, as the USBL system dows not degrade its readings over time and the acoustic modem provides accurate range expansion.

SBL systems require $40m$ baselines for accuracy in deep water [67] and, up to a certain limit, the larger the baseline the better. However, *Smith and Krone* [59] achieved a satisfactory accuracy in a home coming AUV with a 6m portable baseline, constructed with 1.8m collapsable poles in a star configuration. The main goal of this compromise was to be able to rapidly deploy the SBL transceivers to localize the docking station of the AUV.

2.2.2 Long Baseline and GPS Intelligent Buoys

In a Long Baseline or LBL , a series of widely spaced fixed beacons is spread in the area intended to be covered. Each transponder has an unique and distinguishable acoustic signal and the receiver sends an interrogation signal to measure time of flight of each beacon. The sequence of range measurements allows the system to trilaterate the position of the interrogation emitter. Each node of the LBL network must be previously installed and globally referenced. The implementation step can be performed with an auxilary ship [39]. For larger areas or in regions covered with ice, where a ship-based survey would be extremely time consuming, a helicopter can be used to install and georeference each beacon [30]. *Vasilescu et al.* [65] developed an AUV capable of deploying, relocating and recovering the static sensor nodes. Therefore the implementation of a LBL network is not a trivial task and has a non-negligible associated issue, both cost money and time-wise. A basic configuration of a LBL system is illustrated in Figure 2.5.

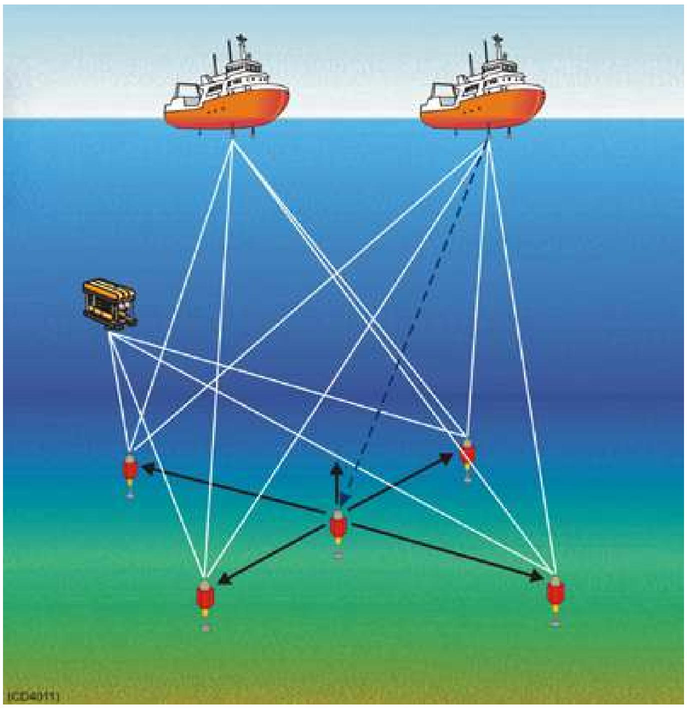


Figure 2.5: Basic configuration of an LBL system. ([47]).

GPS Intelligent Buoys, which are alrealdy georeferenced [2], can replace the fixed beacon in order to attenuate the already mentioned implementation costs. In [71], it was proposed a hierachical system in order to implement a large scale localization system using GPS referenced buoys but with low communication overhead. The authors categorize between anchor nodes and ordinary nodes, in which only the former can communicate with GPS enabled buoys.

Another strategy to reduce implementation costs is to avoid the need of geo-referencing each buoy. This can be accomplished by the use of a self-localizing network, as in [11]. Acoustic modems are applied to allow bidirectional communication between nodes. In addition, self-localizing LBL networks were implemented without the need of additional hardware with a *Simultaneous Localization and Mapping* (SLAM) approach, then solving the localization problem for both transponders and vehicle [46].

2.3 Sonar

Sound travels at a speed of approximately $1500m/s$ in water, what is much slower if compared with light. However, acoustic signals can travel a much further distance before being completely attenuated. Sonar is an acronym for *Sound, Navigation and Ranging* and exploits this characteristic to gather data from the surrounding environment. A passive sonar listens to the environment and has the objective to filter sound signatures of different objects from the background noise. In contrast, an active sonar transmits an acoustic signal and listens to the returning echo. It can determine the distance from the insonified target from time of flight of the reflected signal. The technology was first introduced by Reginald A. Fessenden, driven by the necessity of detecting submerged vehicles during the first World War. The active sonar developed operated with $540Hz$ and had a range of some kilometers [12]. Through the years, technology matured and size and cost of sonar devices allowed its commercial usage in several applications such as obstacle avoidance, seafloor mapping, identification of objects and, of course, localization and navigation of underwater systems.

The operating frequency of an active sonar device dictates its range and how fine grained its response will be. Lower frequencies propagate better through higher density materials and will travel further. High-frequency signals will have a shorter range as a result of its greater attetuation, although it will produce a high detailed output and a good discrimination between targets [9]. Table 2.1 gives a panorama of the different ranges achieved with the increase in frequency and also the associated wavelength.

Frequency	Wavelength	Distance
100 Hz	15 m	1000 km or more
1 kHz	1.5 m	100 km or more
10 kHz	15 cm	10km
25 kHz	6 cm	3km
50 kHz	3 cm	1km
100 kHz	1.5 cm	600m
500 kHz	3 mm	150m
1 MHz	1.5 mm	50m

Table 2.1: Range and frequency relation [9]

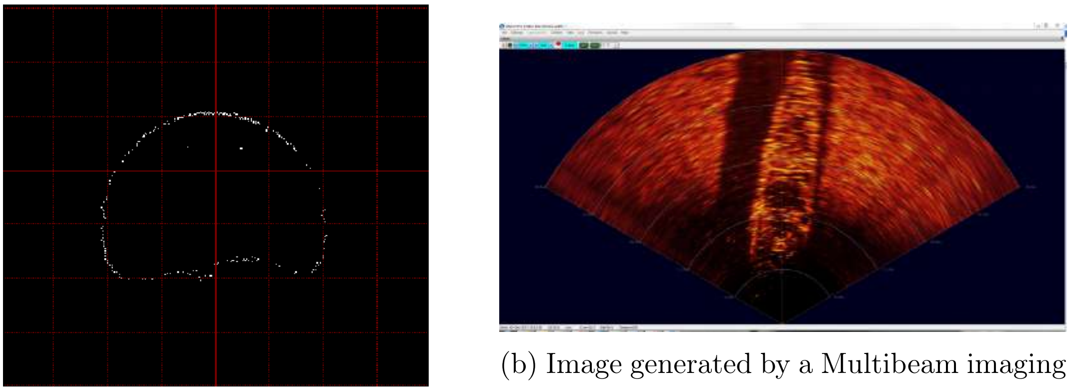
Active sonars and the respective localizations techniques will be covered by the scope of this work and will be the focus point on the proposed method. This type of sonar can be categorized with respect to the data aquisition method, range only or imaging sonars, and with respect to the number of beams, single or multibeam sonars.

2.3.1 Single vs. Multibeam Sonars

Single beam sonars emit one pulse (or beam) and listens to the echo from a single receiving element. Single beam devices can be fixed and point always in the same direction or can be coupled to a motor, which is responsible for changing the bearing of the transducer head. Multibeam sonars explore the concept of wave interference. Combining multiple receiving elements and precisely controlling the timing between the emission and receiving times, multibeam sonar allows to distinguish the direction of the returning echoes [9]. The key advantage of a multibeam sonar is that this kind of device can gather information of a larger area much faster if compared to a single beam sonar. One can consider a multibeam sonar as a colection of single beams, therefore a single beam sonar must have its transducer mechanically moved (rotated or towed) to cover the same insonified area by a multibeam sonar. In addition, for each discretized position the sensor must be still for the whole ping duration, while multibeam sonars can produce over 1000 beams within one cycle.

2.3.2 Ranging vs. Imaging Sonars

Ranging sonars provide only one measurement of range per ping, calculated via TOF. On the other hand, imaging sonars return several intensity measurements, provenient of multiple echoes reflected in the insonified area. These information allows to create an image of the multiple objects present on the emitted signal direction of travel. Figure 2.6 illustrates the difference between an output of a profiling sonar and an imaging sonar.



(a) Ranges output from a profiling sonar

(b) Image generated by a Multibeam imaging sonar

Figure 2.6: Difference between profiling sonar and imaging sonar output.

Ranging Sonars

The beam pattern or beam shape is the first key difference between a ranging and an imaging sonar. Whether it is desirable to have a very focused beam shape or a more wide one splits the two categories apart. A ranging sonar has a conical beam that can be approximated by a pencil beam shape and therefore provides an accurate cross-sectional profile. Figure 2.7 illustrates the pencilar beam shape.

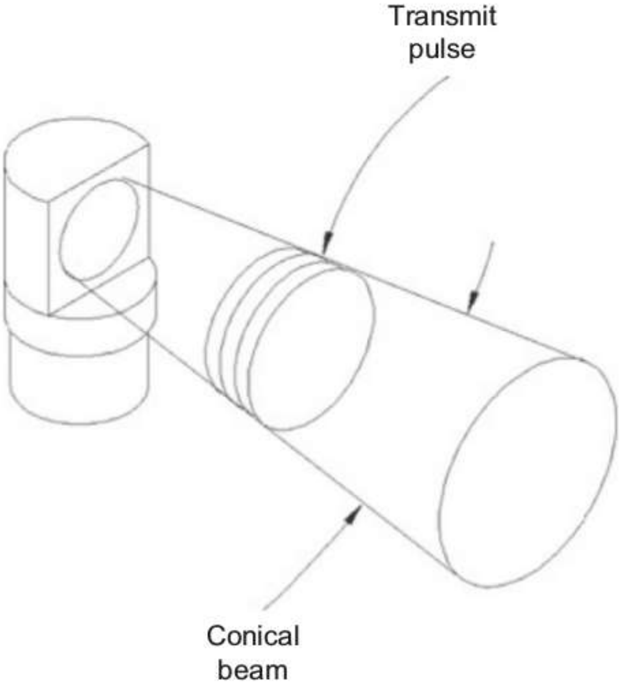
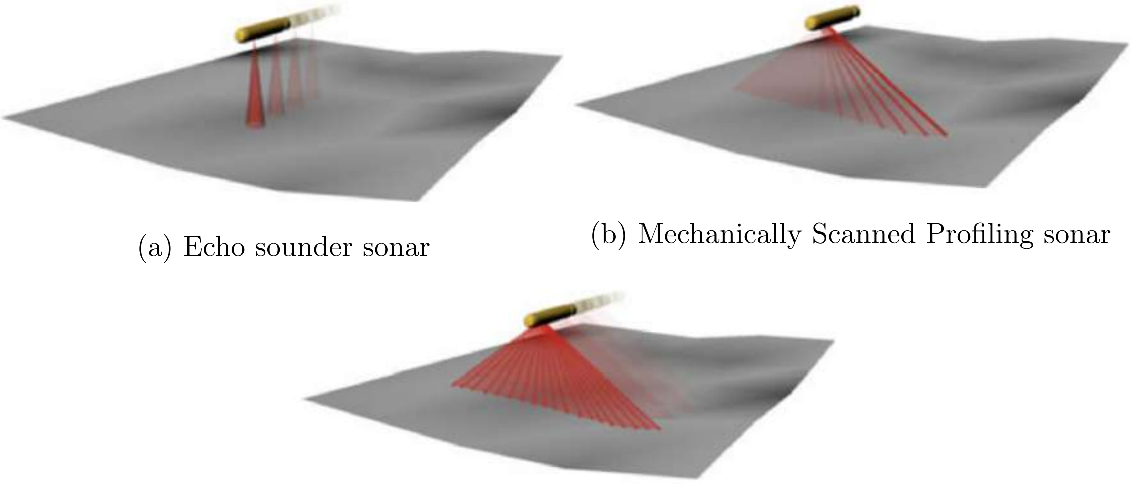


Figure 2.7: Pencil beam shape of a ranging sonar ([9]).

The most simple sonar with this beam pattern is an echo sounder. This device is composed by single fixed transducer, usually mounted towards the seabed. Estimating the distance by TOF, echo-sounders are generally applied as altimeters, measuring the distance of the vehicle to the sea floor. *Fairfield et al.* [17, 19] implemented a SLAM algorithm for an AUV for exploration of Flooded Sinkholes using an array of echo sounders. The vehicle was equiped with 54 sonars displaced around its body and each device has a beam opening of two degrees. Range measurements up to 200m were gathered to form a map of the cenotes of Sistema Zacatón in Tamaulipas, Mexico.

A mechanically scanned profiling sonar follows the same principle of the echo-sounder, although the transducer head is attached to a stepper motor. The sensor can provide up to a 360^o scan, with a beam width as low as 0.45^o. The output of a profiling sensor consists of distance measurements to the first target hit and the respective angular position of the motor. This two measurements together can generate a scan in polar coordinates, what can compare to a profiling sonar with a 2D laserscanner. However, as sounds travels much slower, the time the transducer must wait for the returning signal is much longer and results in a lower sampling rate. Depending on the criteria adopted to consider an echo as a valid hit, the sampling rate can be increased or decreased. If the sensor considers the first echo above a threshold as a valid return, the duration of a single ping will be shorter if compared to considering the peak echo as a detection. Instead of receiving the first hit and moving the transducer head, the sensor must wait a time long enough to allow a possible return coming from an object within the maximum range of the sensor. Profiling sonars are applied in localization and mapping of underwater vehicle analogously to 2D lasescanners, taking into account the lower sampling rate and poorer accuracy. In [40],a SLAM navigation method was proposed to map a large seafloor area using artificially deployed landmarks as vertical pole-like acoustic reflectors. *Maurelli et al.* demonstrated a localization system without the need to deploy any artificial infrastructure and proposed a featureless particle filter to localize an AUV [44, 45].

To overcome the slow scan times, a multibeam echo sounder or multibeam profiler can be appplied. Instead of only one transducer, this kind of sensor is composed of an array of hydrophones and are capable of measuring a whole 2D scan sector simultaneously. This kind of sensors are applied in the construction of bathymetric maps. Figure 2.8 illustrates the three types of range-only sonar. *Barkby et al.* presented a solution to the SLAM problem of constructing bathymetric maps using a particle filter. A multibeam sonar with 120 beams, with 3 degrees opening each, was responsible for gathering the profile of the seafloor and the algorithm was sucessful to handle large areas, even with little or no overlap in the mapping process. In [18], *Fairfield et al.* proposed a similar method from [17, 19], however an active localization strategy was used to implement a mapping of large featureless area of ocean floor. The 256×1^o beams spread in a downward facing 150^o fan were deliberately moved to regions that are predicted to reduce the vehicle's position uncertainty and therefore improved the convergence and accuracy of the whole method.



(a) Echo sounder sonar

(b) Mechanically Scanned Profiling sonar

(c) Multibeam Profiler Sonar

Figure 2.8: Most common range-only sonars (adapted from [52]).

Imaging Sonars

As exemplified in Figure 2.6, an imaging sonar produces an image of the insonified area, similar to the operation of a radar. However, in contrast with the highly focused pencilar beam of ranging sonars, an imaging sonar has a fan shaped beam, as illustrated in Fig. 2.9. This format, with a wide vertical beamwidth and a small horizontal opening, is well suited to perform obstacle avoidance and navigation. Objects or targets present above or bellow the transducer axis will be detected, while maintaning a good horizontal angular resolution. The Tritech Micron, a common choice for imaging sonar in small ROVs and AUVs given its low weight and affordable price (LUMA, ROSA and Flatfish all have one unit), has a vertical beamwidth of 35^o while the horizontal opening is of only 3^o.

As the emitted pulse covers a greater area than in the ranging sonars case, multiple returning echoes, reflected from different distances, may arrive within a single ping. Therefore, instead of returning a single ranging value, imaging sonars return an array of every reflection above a threshold. To do so, the sonar listens to the returning echoes in pre-defined periods of time for each acoustic signal transmited. This approach allows the imaging sonar to associate each reflection to a particular time of arrival. In other words, each element of the output array corresponds to a

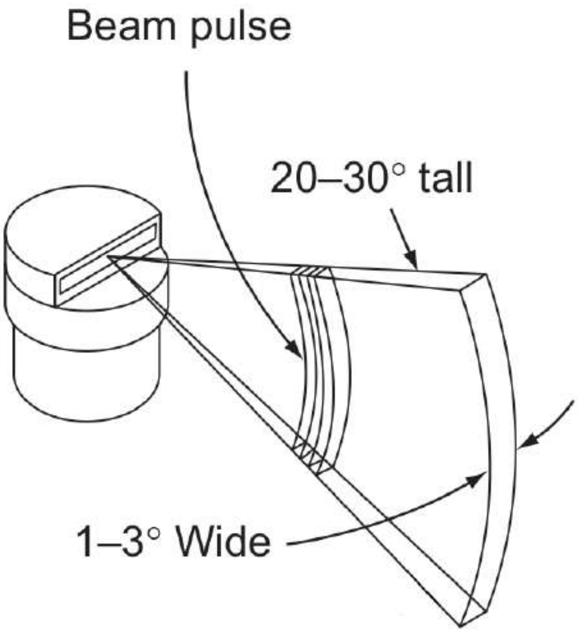
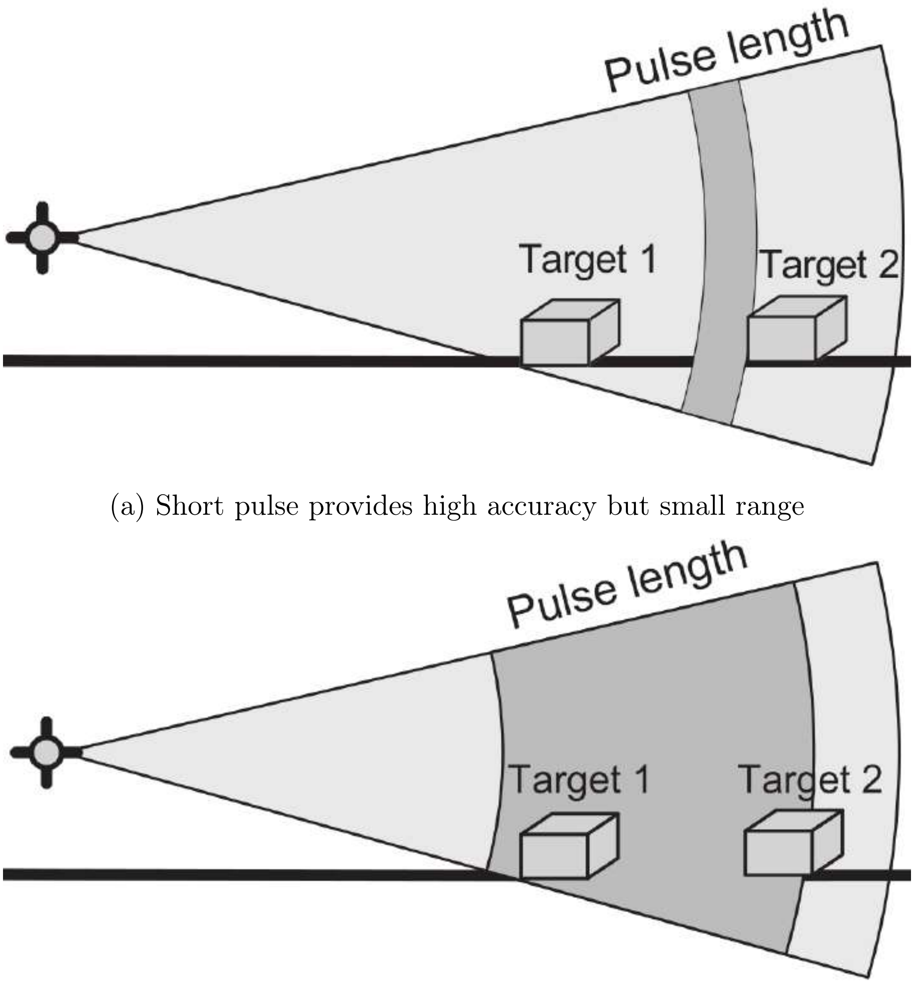


Figure 2.9: Fan shaped Beam the associated bin readings representation ([9]).

specific distance from the transducer head. By varying the length of the listening period, the sensor can achieve a higher resolution along the beam axis or a greater range, as the total number of discretized spaces tends to be fixed on the sonar device (limited sized array message). This discretized time period (or equidistant regions) are illustrated in Figure 2.10.

Another particular characteristic introduced by the fan shaped beam is that portions of the acoustic signal may hit targets with a greater angle of incidence if compared to the almost orthogonal incident pencilar beam from a ranging sonar. This means that the intensity of the returning echoes will vary depending on how perpendicular the target is to the transducer beam axis. In conclusion, to be able to show the nuances between objects in the image, the output of an imaging sonar is an array of intensities of the returning echoes. The beam is discretized along the transducer axis and each element is called *bin*. Each bin is associated to a distance with the respective TOF of that particular region of space. The top part of Figure 2.11 illustrates the intensities received by an imaging sonar in the scenario illustrated by its bottom part. It is possible to visualize the backscattered signal from the seabed being much weaker than the high intensity return from the bump on the floor. Most of the imaging sonar have a *Time Varyng Gain* (TVG) to compensate the attetuantion due the distance travelled within maximum range.

Figure 2.11 also allows the reader to visualize an important phenomenom which is fundamental to the interpretation of a sonar image: shadows. The region behind an hit target is not insonified and therefore cannot produce any echo. This can be perceived as a region of no intensity reading after a high peak return. Sonars opera-



(a) Short pulse provides high accuracy but small range

(b) Long pulse provides a longer range but compromises accuracy

Figure 2.10: Comparison between a short and long pulse length and its influency in the accuracy along the transducer axis (adapted from [9]).

tors are trained to identify objects by their shadows as they can produce much more semantic information than their reflections, what is exemplified in Figure 2.12. The height of the object responsible by a strong reflection can be inferred by the length of the shadows casted on the image. Object identification and tracking methods based on the analysis of the image produced by an imaging sonar also implement this strategy [48].

The wide vertical opening is particularly useful for detecting the presence of an object in the insonified area, which is why this type of sonar is often used in obstacle avoidance. However, it also introduces an indetermination of the vertical position, as can be seen in Figure 2.13. Due to this indetermination, many authors consider that only a 2D projection can be inferred and all 3D information is lost in this process. This assumption tends to simplify a imaging sonar as a very noisy range sensor, as it will be discussed in the Section 2.3.3.

Within this category of sonar, considering the beam shape as a comparison pa-

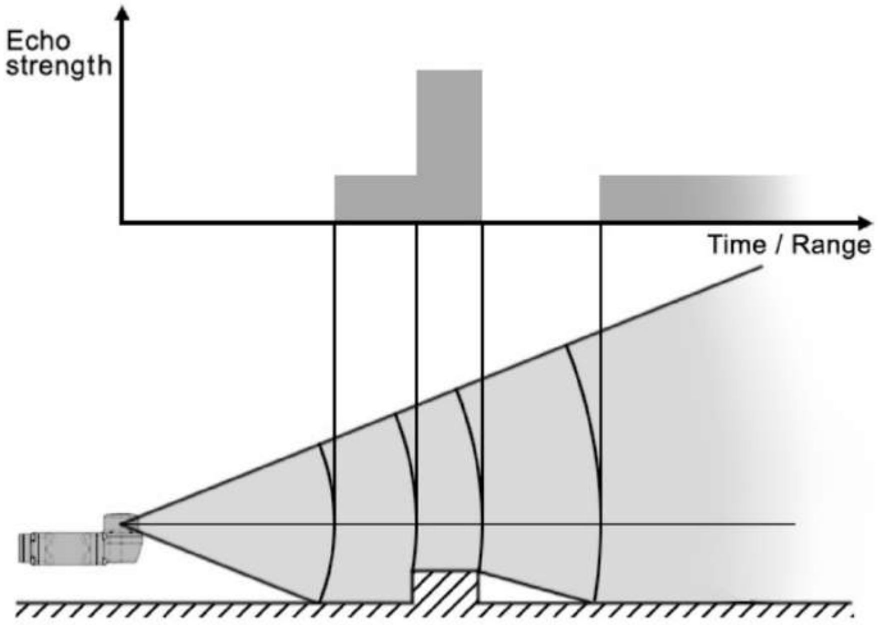


Figure 2.11: Example of intensity readings from a imaging sonar ([52]).

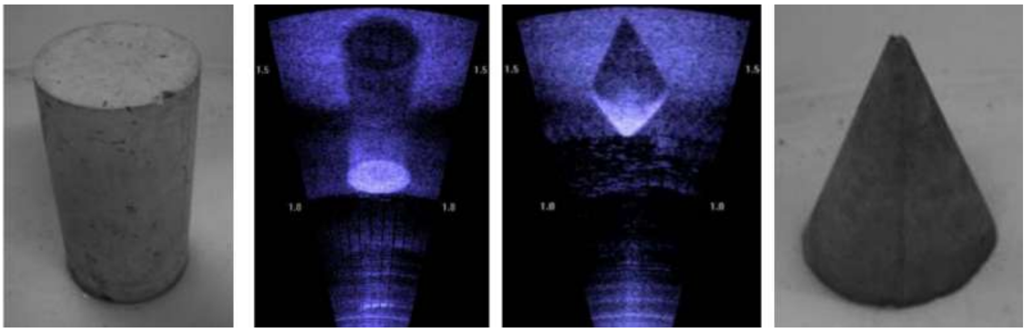


Figure 2.12: Sonar images geneарted by a FLS sonar and their respective and indentifiable shadows (adapted from [22]).

rameter, the most simple is the Side Scan sonar and its fixed fan shaped beam. A Side Scan sonar is designed for imaging large seabed areas, usually on marine surveys and mapping. The sonar is usually towed by a ship or submarine and is faced towards the seafloor. Its wide angle fan shaped beam produces intensity image slices that are stitched togheter along the direction of motion. The towing operation and the downfacing orientation of the sensor make this type of sonar sensitive to the trajectory realized by the vehicle. As there is no overlap between two consecutive readings, the sensor is dependent on multipass missions and there must be an overlaping between the readings from adjacent tracks. With this constraint in mind, *Ruiz et. al* proposed a landmark based SLAM algorithm using a sidescan sonar [57], while *Maurice et al.* proposed a hybrid system, which integrates global range constraints, coming from GPS position of a surface vehicle or buoy and communicated to the system via acoustic modems, to a side-scan sonar [20].

A *Mechanically Scanned Imaging Sonar*(MSIS) is analogous to a mechanically

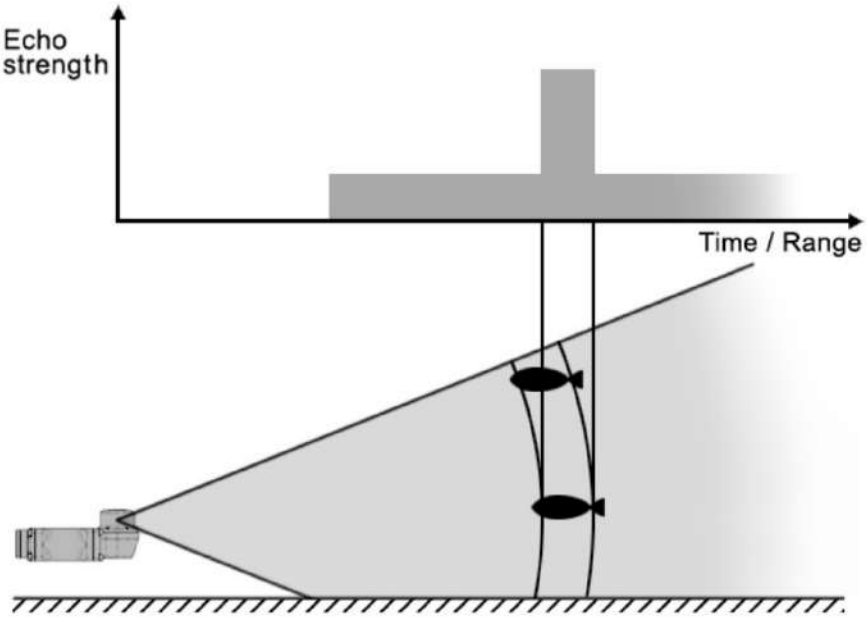
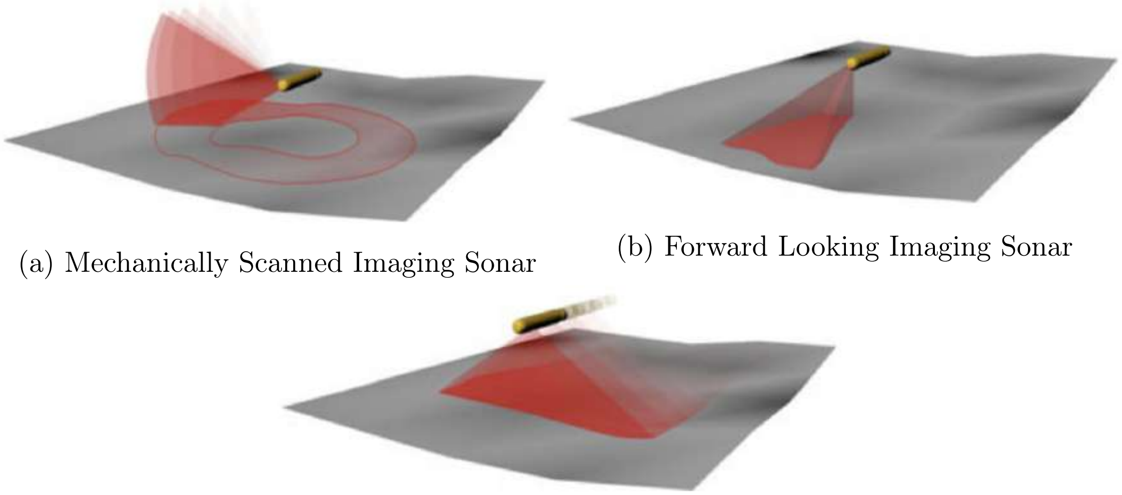


Figure 2.13: Ambiguity in the vertical position ([52]).

scanned profiler and have a stepper motor to swipe the transducer head. The fan shaped beam is then rotated up to 360^o to gather a full scan. As in the profiler sensor, the ping period has a direct impact in the sampling rate. If the bin length is set to increase range, the sensor must wait longer to move to the next bearing position.



(a) Mechanically Scanned Imaging Sonar

(b) Forward Looking Imaging Sonar

(c) Sidescan Sonar

Figure 2.14: Most common active sonar types (adapted from [52]).

MSIS devices are being used in relatively small AUVs thanks to its small size, low weight and low cost. Other relevant aspect, however, is that they have comparatively low resolutions and a very low sampling rates. It has therefore been mainly applied in the localization of underwater vehicles in small areas and structured or partially structured environments with features easy to detect. In [53], a MSIS device is used in a AUV developed for the inspection of hydroelectric dams and it was proposed a SLAM algorithm based in the extraction of line features of planar structures of artificially made environments. As the sampling rate of the sonar is very low, the vehicle cannot be considered to stand still during a full scan and the distortion caused by the vehicle's motion had to be corrected based on the information from the dead reckoning system. The full 360^o scan is considered as a 2D plane reading and each beam is transformed into a single distance measurement by thresholding. However the fan shaped beam has a wide horizontal opening if compared to a laserscanner for example. To overcome this limitation it was proposed, in [54, 55], an extended sonar model taking into account the horizontal beam width and dealing with the uncertainties on the incidence angle.

As an alternative to the feature extraction of the environment, *Hernandez et al.* [25] proposed a probabilistic scan matching algorithm using a MSIS device. This technique estimates the robot relative displacement between two configurations by maximizing the overlap between two range scans. The MSISpIC, which is an extension of the pIC method (*probabilistic Iterative Correspondence*) specific for MSIS sensors, aims to take into account the noisy and sparse measurements obtained from a MSIS in order to better estimate the displacement between scans. This approach also uses the data from a dead reckoning to compensate the motion of the robot during the scans. Later, *Mallios et al.* extends the MSISpIC approach by maintaining each new pose of a scan on a second *Augmented State Extended Kalman Filter* (ASEKF), and compare the scans with previous scans that are in the nearby area [41]. *Maurelli et al.* implemented the same algorithm used for a mechanically scanned profilling sonar with an MSIS device, raising satisfatory results in a 2.5D environment. However, it was not clear how the vertical ambiguity was dealt by the authors.

Finally, a *Multibeam Forward Imaging Sonar* or *Foward-Looking Sonar* (FLS) uses an array of receivers to identify the direction of the returning echoes and can scan a larger horizontal sector per ping. The Tritech Gemini 720i, used in the FlatFish AUV scans a 120^o sector composed by 256 beams. A FLS can achieve high resolution (8mm in range for the Gemini), however the prices can be also very high, specially if compared to a MSIS. Figure 2.15 illustrates the beam shape of a FLS and how the environment is insonified. As it single beam counterpart, a multibeam forward looking sonar is primarly used to obstacle avoidance and vertical features dectetion.

Karoui et al. proposed an automatic sea-surface object detection analysing FLS images sequences and it was capable to detect and track man made objects, such as buoys, boats and vessels. The method was based on the sound signature imprinted in the sonar output [36]. In [68], a undersea inspection of ship hull is implemented using a hovering AUV equipped with a FLS sonar. The sonar was a DIDSON with a vertical beamwidth of 12^o and 28.8^o of horizontal FOV, in which 512 beams are uniformly spaced. The localization and mapping algorithm was based on the identification of deployed targets which position was previously measured by a team of divers. *Johannsson et al.* [32] proposed a method for ship hull inspection and harbor surveillance using the same AUV platform, but instead of relying on an auxiliary infrastructure (deployed landmarks), the authors extracted natural features based on sharp transitions on the sonar image. In [28], *Hurtos et al.* proposed a Fourier-Based featureless mosaicing method, which analysed the phase correlation between two consecutive sonar images, later it achivied real time processing using the same method [29].

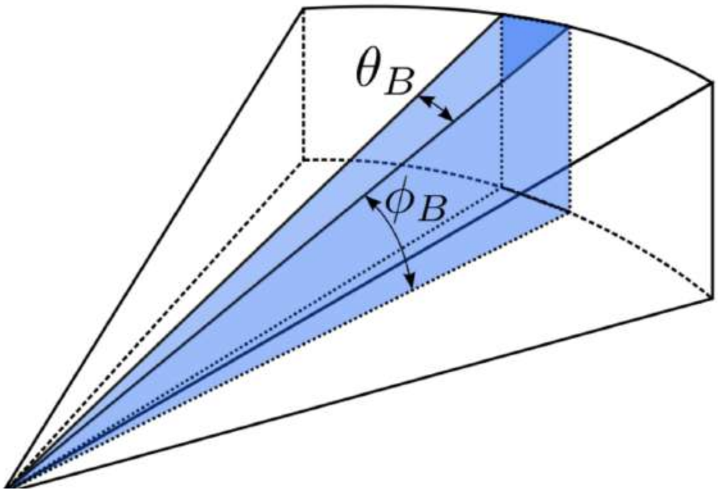


Figure 2.15: Beam shape of a FLS and the representation of a horizontal beam discretization ([15]).

2.3.3 Beam segmentation and range extraction

All sonar based techniques described have the necessity to evaluate the incoming sonar beams or scans with some reference, which allows the system to observe some portion of the environment and infer its position. Profiling sonars are modelled as range finder sensors and its beam forming process, which aims to achieve a very narrow penciclar shaped beam, allows the direct analogy with the well mature laser based localization theory [31]. The returning echo TOF is converted to a distance measurement based on the speed of sound in water and it is compared to a reference distance, usually obtained in a ray casting process. This process simulates an unidimensional ray, which travels from the origin of the sensor until it intersecs with some object in the map that represents the environment (as illustrated in Figure 2.16). Due to limitations of producing a colimated sound “ray”, the angular precision of a sonar will always be much lower if compared to a LIDAR (*Light Detection and Ranging*) system.

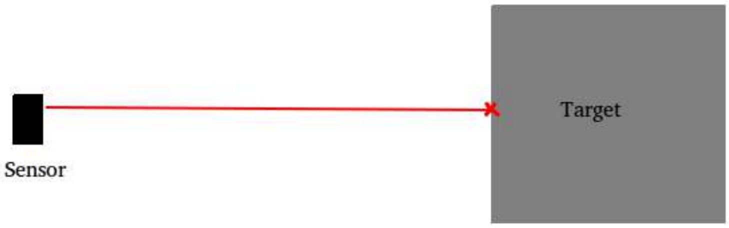


Figure 2.16: Horizontal ray casting from a sensor to a target.

Some authors incorporated this limitations into localization and or mapping process, as for instance, *Fairfield et. al* represented the pencilar beam shaped as a narrow cone, which was constructed as a bundle of rays casted from the sonar head transducer origin. This concept is illustrated in Figure 2.17, which the cone model is represented in an Octree map, a type of grid that divides the space in octants to increase memory efficiency [27]. This representantion was implemented with an echo sounder sonar in [19] and also with a multibeam profiler [18]. A pre-computation approach can be implemented to cope with the high processing power demand, however the memory requirements of a lookup table can be significant depending on the size and resolution of map.

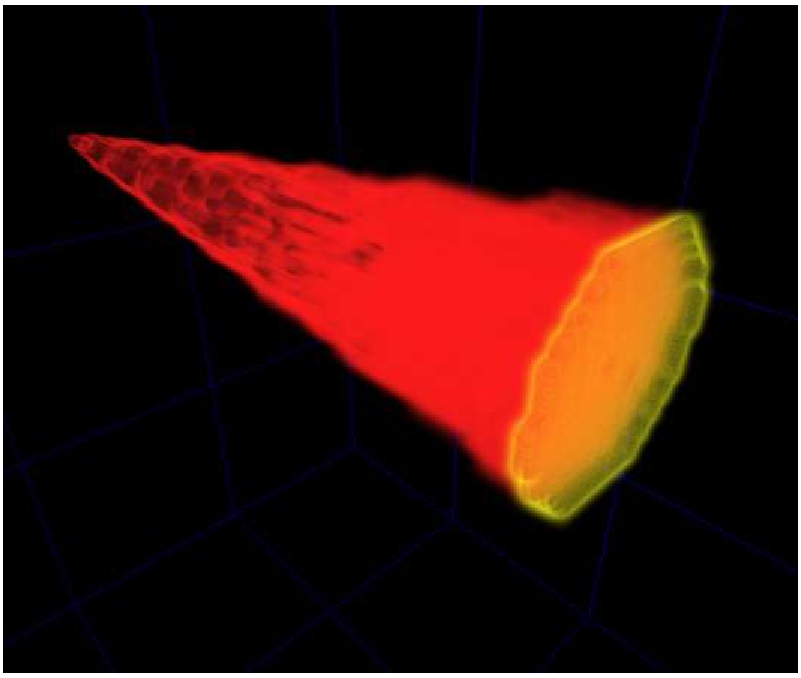


Figure 2.17: Cone model used by *Fairfield et al.* [18] in opposition to a ideal ray model.

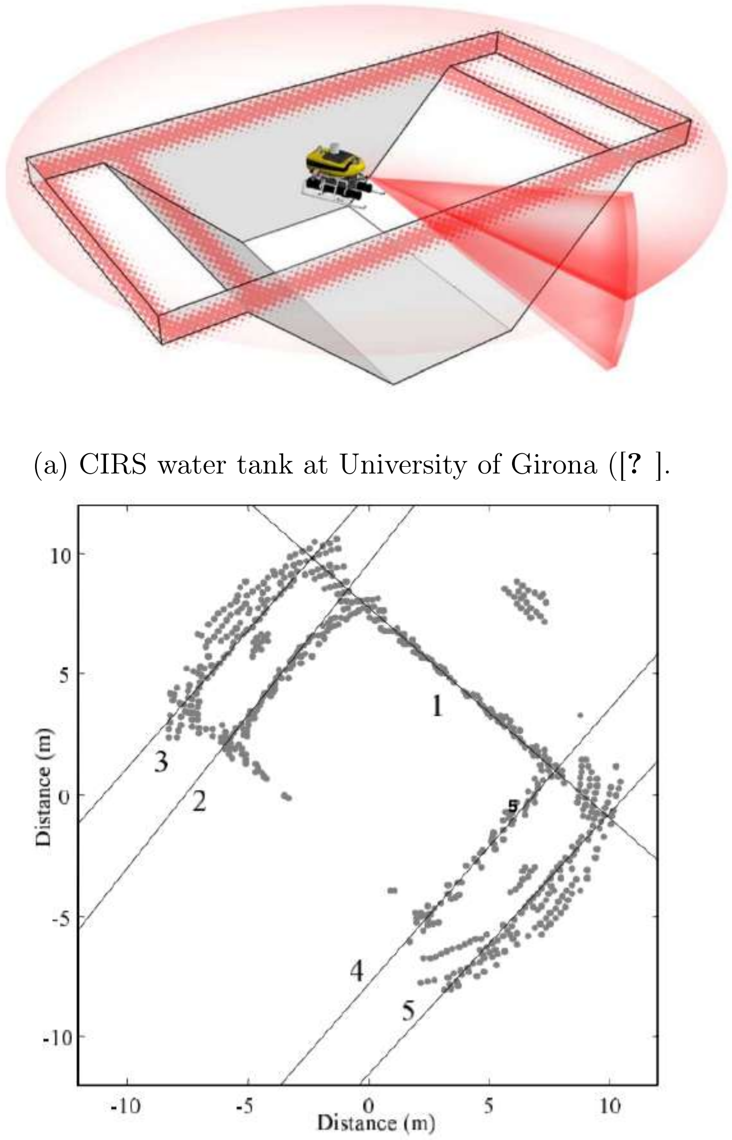
The approximation of a profiling sonar as a pure range finder sensor, even though not optimal, has shown to be efficient as discussed in Section 2.3.2. However, a imaging sonar demands some data processing to be able to provide a range value.

In order to extract a range measurement from an imaging sonar beam, the intensity profile is processed with the goal of extracting its local maxima, where the high peak intensity values are considered as returning echoes from significant targets. In [53], it was proposed a segmentation method that first filters out the background noise using simply a pre-defined thereshold. Then, the algorithm searches for the local maxima in the intensity array. The range measurements associated with peak intensity bins are then taken into account, discarding the intensity value. It is worth noting that this segmentation criteria can handle more than one distance reading per beam, what is an expansion of the basic range finder sensor representation. In a following paper [54], a refinement on the beam segmentation is proposed and a minimum distance between two local maxima is adopted. This criteria tries to avoid the computation of different range measurements for a same object. The same segmentation method is applied in [55]. Dinamically thresholding can be applied to filter the background noise and *Heidarsson et al.* [24] proposed a histogram analysis of the intensities of each beam array. A threshold is set between the first two intensity value modes, assuming that the first mode contains only low strengh echoes from background noise.

Once the beam segmentation process is concluded, a range reference must be determined. In the particular approach proposed by *Ribas et al.* [53], reference ranges are geometric defined on a line-feature map, as illustrated in Figure 2.18b. This approach cannot represent the vertical position of this lines, what means that the representation of the environement is limited to 2D maps and the lines are "infinite walls", what in the test scenario of the platform is already a simplification that not fully represents the world (Figure 2.18a).

Following the same segmentation method, a probabilistic scan matching technique is applied to estimate the displacement between two scans[25, 41–43]. Either the feature or the featureless approach uses a full scan as measurement, what means that both need to cope with the slow sampling rate presented by a mechanically scanned imaging sonar. The sonar head cannot be assumed to stand still and remains at the same position during the whole scan and the scan distortion induced by the movement needs to be corrected, as illustrated in Figure 2.19.

In addition, [44, 45, 50] have shown that is possible to use a similar approach to localize an underwater vehicle with a MSIS, analysing only single beams at a time. The authors implemented a particle filter and for each particle they compare the range measurements filtered from the real sonar beam r with a simulated reading s. The simulated array s was computed by ray-tracing algorithm and the authors exchange a profiling sonar and MSIS sonar without expliciting any change in the



(a) CIRS water tank at University of Girona ([**?**].

(b) Multiple line intersection along the way of travel (adapted from[53]).

Figure 2.18: Water tank test sceneario at University of Girona and the filtered readings generated by a MSIS.

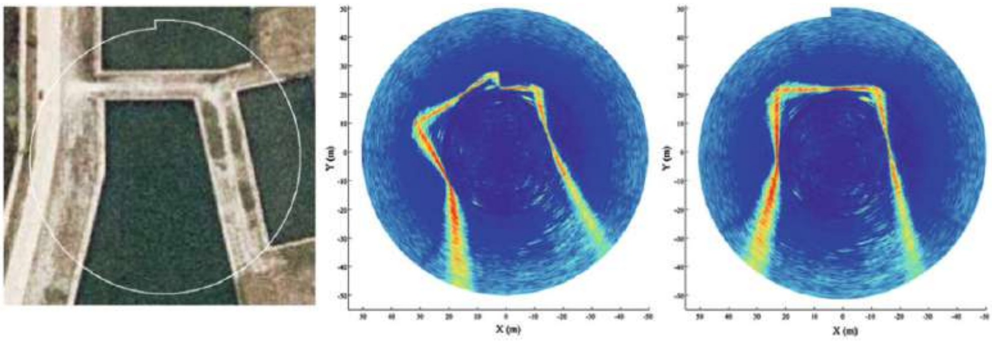


Figure 2.19: Motion induced distortion correction ([43]).

proposed method. For that reason it was assumed that only a horizontal beam is considered and the vertical opening of the sonar is not taken into account on the mentioned papers.

Even though some authors incorporated intrisec and specific sonar characteristics, as the representation of a profilling sonar beam as a narrow cone [18] and the incorporation of backscattering properties to better deal with the narrow horizontal opening of imaging sonars [53–55]. All the presented methods so far do not cope with the wide vertical beamwidth and make simplifications on the principle of operations of an imaging sonar to represent it as a range finder sensor. This simplifications reflect on the environment representation and in how the reference or observation of the state can be determined. The first major assumption considers that the vertical ambiguity introduced by the wide vertical beamwidth, presented in the Section 2.3.2, eliminates all 3D information and limits the imaging sonar to a 2D sensor. For that reason, the applications of the imaging sonar sensor for localization, in the author knowledge, are currently limited to 2.5D maps. The reference range cannot be determined in 3D complex environments with the geometric or ray casting approach, as the wide vertical opening is not represented in these methods. In addition, when an imaging sonar is modelled as a range finder sensor, the intensity value of each returning echo becomes meaningless and it is discarded. Therefore, a more detailed sensor model, which can simulate the behavior of wide beam shape and its propagation, could explore the currently unused information provided by the devices to deal with more complex environments and converge more efficiently.

Chapter 3

Underwater Localization with an Imaging Sonar

To deal with the localization problem of underwater systems, acoustic signals are best suited as it is not rapidly attenuated as high frequency ones. Acoustic beacons requires high implementation and operations costs, with the use of support vessels. Therefore, for small AUVs and underwater systems, the use of sonar devices to gather information about the environment is well suited, as they provide flexibility to operate in different locations without the need of any additional infraestructure. Imaging sonar plays an important role in this scenario as its wide vertical beam allows a good obstacle avoidance capability.

This work proposes a localization method using a mechanically scanned imaging sonar (MSIS), which stands out by its low cost and weight. The proposed method implements a Particle Filter, a Bayesian Estimator, and introduces a measurement model based on sonar simulation theory. To the best of author's knowledge, there is no similar approach in the literature, as the current sonar simulation methods target synthetic data generation for object recognition. This stands as the major contribution of this work and allows the representation of the full beam shape of imaging sonars and the computation of intensity values. This detailment in the representation allows the algorithm to estimate the position of underwater systems in complex three dimensional environments and also to extract information outside the main axis of the sensor. To understand the proposed solution key concepts of recursive and stochastic state estimation must be introduced. The material presented in Section 3.1 is mainly based on [63].

3.1 Recursive and Stochastic state estimation

A model-based paradigm, predominant in the 1970s, assumes a fully modeled and deterministic robot and environment [62]. The space-state model of a system (or robot) encodes all the relevant information to describe its behavior through time. Although, recent robotics aplications, specially in the field of mobile robots, operate in much more unstructured and dynamic environments. The level of uncertainty that this kind of system is exposed is much higher if compared to *"classic"* robotics applications as assembly lines, which perform the exact same task exaustively. The drift caused by an unmodelled underwater current, exact weight distribution and the noisy readings coming from a sonar sensor are examples of uncertainties that underwater robots are exposed. Therefore, the assumption that the behavior of the system is fully modeled and its state estimation is correct needs to be reevaluated.

Hence to cope with this kind of applications, the main idea of probabilistic (or stochastic) state estimation is to explicity model the underlying uncertainties. Instead of estimating the state in a deterministic way, considering that the model is exact, the probabilistic paradigm estimates the true state using probability distributions. In other words, the main goal is to estimate how likely to be true the estimation is.

In order to present the basic theory of *stochastic state estimation*, it is important to first introduce some concepts.

The state transition function that models the behavior and evolution of the estimated state during time cannot be represented by a deterministic function. Instead, the concept of a *state transition probability* must be introduced:

$$p(x_t|x_{0:t-1}, u_{1:t}, z_{0:t-1}). \tag{3.1}$$

where it defines all the possible new states that can be reached given the previous states, control inputs and measurements[1]. However, one can assume the *completeness* of state x_t and that it satisfies the Markov property, i.e the current state encodes all the information about the previous states and control inputs. Therefore, all future conditional probabilities depend only on the current state and do not rely on any of the past states $x_{0:t-1}$, control inputs $u_{0:t-1}$ or measurements $z_{0:t-1}$. It is possible to represent Eq. 3.1 as being conditioned only to the previous state x_{t-1} and the last control input u_t:

$$p(x_t|x_{0:t-1}, u_{1:t}, z_{0:t-1}) = p(x_t|x_{t-1}, u_t). \tag{3.2}$$

[1] The same convention from [63] is adopted, it is considered that the robot or the particular system executes the control input u_t before it performs the latest measurement z_t.

Once the uncertainties are explicity taken into account on the process system modelling, the prediction the state evolution, introduced in 3.2, will accumulate the errors and will become meaningless over time. Unless it is grounded on observations of the actual state. The process, in which measurements z_t are generated, can be also expressed by a conditional probability distribution, called *measurement probability* distribution. Analogously with the state transition probability, and considereing the completeness of the state it can be defined as:

$$p(z_t|x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(z_t|x_t). \tag{3.3}$$

This probability distribution defines how likely the measurement z_t is to happend, given the state x_t. This distribution is not always trivial to be defined and will play a fundamental role in the development of this work. On top of that, rarely the state of a system can be measured directly. For that reason, it is defined as the *internal belief* $bel(x_t)$ of the system, all the information available to the robot about the true state and the environment at a given time t:

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}). \tag{3.4}$$

Eq. 3.4 is a probability distribution over the state x_t at time t, conditioned on all past measurements $z_{1:t}$ and all past controls $u_{1:t}$. It represents the knowledge of the system about its actual state and assigns a probability (or density value) to each possible hypothesis of the true state.

3.1.1 General Bayes Filter

The Bayesian filtering is the base of probabilistic robotics and relies on the Bayes Rule to maintain the internal belief of the system recursively. The Bayes Rules allows one to infer a quantity x from the available data y:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}. \tag{3.5}$$

The probability p(x | y) is called the posterior probability distribution over the random variable X. In robotics the posterior distribution is defined by the *belief*, introduced by Equation 3.4. The prior probability distribution summarizes the knowledge of the system prior the incorporation of the latest measurement z_t and is denoted $\overline{bel}(x_t)$:

$$\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t}). \tag{3.6}$$

Given an initial state estimate x_{t_0}, the BF calculates the posterior recursively in two steps: a prediction step and a correction step. The first step of the Bayes filter is responsible for incorporating the control input u_t to the previous belief of the system $bel(x_{t-1})$. Expanding the prior distribution $\overline{bel}(x_t)$ using the *theorem of total probability*, it can be written as:

$$\begin{aligned}\overline{bel}(x_t) &= \int p(x_t|x_{t-1}, u_t)p(x_{t-1}|z_{1:t-1}, u_{1:t-1})dx_{t-1} \\ &= \int p(x_t|x_{t-1}, u_t)bel(x_{t-1})dx_{t-1}.\end{aligned} \tag{3.7}$$

The prediction step is then the computation of the prior internal belief of the system $\overline{bel}(x_t)$ expressed as a convolution of posterior belief at time $t-1$ with the state transition probability, that describes the influence of the lastest control input u_t over the last state x_{t-1}. With the arrival of the lastest sensor measurements, the prediction is corrected in the second step. This can be explained based on the Bayes Rule (Equation 3.5) applied to the posterior as follows:

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) = \frac{p(z_t|x_t, z_{1:t-1}, u_{1:t})p(x_t|z_{1:t-1}, u_{1:t})}{p(z_t|z_{1:t}, u_{1:t})}. \tag{3.8}$$

Once again exploring the completeness of the state x_t, the history of previous measurements and controls provide no additional information and Eq. 3.8 can be simplified as:

$$bel(x_t) = \frac{p(z_t|x_t, z_{1:t-1}, u_{1:t})p(x_t|z_{1:t-1}, u_{1:t})}{p(z_t|z_{1:t}, u_{1:t})} \overset{\text{Markov}}{=} \frac{p(z_t|x_t)p(x_t|z_{1:t-1}, u_{1:t})}{p(z_t|z_{1:t}, u_{1:t})}. \tag{3.9}$$

One may recognize the measurement probabilty (Eq. 3.3) as the first term of the numerator, and the prior $\overline{bel}(x_t)$ as the second term. Due the fact that the denominator in Eq 3.9 doesn't depend on the state x_t, it will be constant to any value of x_t and can be replaced by a normalizer variable η:

$$bel(x_t) = \eta p(z_t|x_t)\overline{bel}(x_t). \tag{3.10}$$

In conclusion, the correction step computes the current posterior *belief* by multiplying the prior $\overline{bel}(x_t)$ by the probability that the measurement z_t may have been observed.

The Bayes Filter can be summarized as in Algorithm 1 and it is possible to visualize that given an intial internal belief $bel(x_{t_0})$, i.e an initial guess, the Bayes Filter can recursively update the internal state estimation of the system based on the incoming control inputs and measurements. Even though the filter is robust to unmodeled dynamics, inaccuracies in the probabilistic models, and approximation errors [63], the algorithm requires the exact calculation of the probability distributions presented at each step. In practice, there are, except for very simple estimation problems, no closed solutions for the integral in line 3 (Eq. 3.7) and the product in line 4 (Eq. 3.10). The most common practical implementations of the Bayes Filter will be presented in the rest of this section.

Algorithm 1 Bayes Filter Pseudocode

```
   BayesFilter($bel(x_{t-1}, u_t, z_t)$)
1: for all $x_t$ do
2:     $\overline{bel}(x_t) = \int p(x_t|x_{t-1}, u_t) bel(x_{t-1}) dx_{t-1}$
3:     $bel(x_t) = \eta p(z_t|x_t) p(x_t|z_{1:t-1}, u_{1:t})$
   return $bel(x_t)$
```

3.1.2 Kalman Filter

The Kalman Filter (KF) [35] relies on some important assumptions and simplifications of the general Bayes Filter to be able to introduce a practical implementation to real world problems. The first assumption is that the beliefs are assumed to be Multivariate Normal Distributions(MVN), described by its two moments, the mean μ_t and covariance Σ_t.

$$bel(x_t) = \mathcal{N}(x_t, \mu_t, \Sigma_t) = \frac{exp\{(x_t - \mu_t)^T \Sigma_t^{-1} (x_t - \mu_t)\}}{\sqrt{det 2\pi \Sigma_t}}. \tag{3.11}$$

To this assumption holds, the state transition and measurement probability funtions (Equations 3.1 and 3.3) must be linear functions in its arguments, with a zero mean Gaussian noise associated:

$$x_t = A x_{t-1} + B u_t + \epsilon_t, \tag{3.12}$$

$$z_t = C x_t + \delta_t, \tag{3.13}$$

where A is a square matrix of size $n \times n$, respecting the state vector x_t dimensions, and B is $n \times m$, where m is the dimension of the control input vector u_t. Finally C is a $k \times n$ matrix with k being the dimension of the measurement vector z_t. The randomness of the state transition and the measurement noise are represented by the addition of the gaussian noise vectors ϵ_t and δ_t, which have zero mean and known covariance R and Q, respectively. As Eq. 3.12 and 3.13 are linear in its arguments with additive Gaussian noise, the state transition and measurerent probabilities are defined as MVN:

$$p(x_t|x_{t-1}, u_t) = \frac{exp\{-\frac{1}{2}(x_t - A x_{t-1} - B u_t)^T R^{-1} (x_t - A x_{t-1} - B u_t)\}}{\sqrt{det 2\pi R}}, \tag{3.14}$$

$$p(z_t|x_t) = \frac{exp\{(-\frac{1}{2}z_t - Cx_t)^T Q^{-1}(z_t - Cx_t)\}}{\sqrt{det2\pi Q}}. \tag{3.15}$$

To ensure that the posterior is always a Gaussian, one more assumption needs to be made and the initial belief $bel(x_0)$ must also be normal distributed as

$$bel(x_0) = \frac{exp\{(x_0 - \mu_0)^T \Sigma_0^{-1}(x_0 - \mu_0)\}}{\sqrt{det2\pi\Sigma_0}}. \tag{3.16}$$

As the Bayes Filter, the KF has basically a prediction and a correction step. The prediction step calculates $\overline{bel}(x_t)$, which is normal distributed and represented by the two moments $\overline{\mu}_t$ and $\overline{\Sigma}_t$ by incorporating the latest control u_t. The mean is updated using the state transition function 3.12:

$$\overline{\mu}_t = A_t\mu_{t-1} + B_t u_t, \tag{3.17}$$

$$\overline{\Sigma}_t = A_t\Sigma_{t-1}A_t^T + R. \tag{3.18}$$

In addition, the prior covariance $\overline{\Sigma}_t$ is updated considering that the current state depends on the previous one through the linear matrix A and the contribution of the Gaussian noise is introduced by R. The formal justification of the KF is extensive and will be kept outside the scope of this work without any loss in compreension. The reader can look at [35, 63] for a complete formal mathematical derivation.

The correction step is balanced by the introduction of the Kalman Gain K_t, which acts as a blending factor and specifies how much weight the measurement z_t should have in the correction of the posterior mean and covariance. The KF defines its gain K_t in order to minimize the posterior covariance error [70] and is calculated as follows:

$$K_t = \overline{\Sigma}_t C_t^T(C_t\overline{\Sigma}_t C_t^T + Q)^{-1}. \tag{3.19}$$

Once defined the Kalman gain for the time t, the correction step can be performed and the posterior mean μ_t and covariance Σ_t can be updated.

$$\mu_t = \overline{\mu}_t + K_t(z_t - C_t\overline{\mu}_t), \tag{3.20}$$

$$\Sigma_t = (I - K_tC_t)\overline{\Sigma}_t. \tag{3.21}$$

One of the key advantages of the Kalman Filter is its computational efficiency. The complexity of the algorithm is lower bounded by the matrix inversion operations involved in the calculation of the Kalman Gain, which is O($k^{2.8}$) and k is the dimension of the measurement vector z_t. And the matrixes multiplication imposes a complexity of order O(n^2), with n being the number of states. Depending on the applications one term can dominate the other, e.g in mapping application, where the state space can have hundreds up to thousands dimensions, the dominant term is the complexity O(n^2).

It is important to note that the Kalman Filter constraints narrow the state space representation to linear systems with a Gaussian noise, what make this approach limited to more simple models and unimodal distributions. Applications with multiple hypothesis, such as global estimation, are not well suited to be implemented with a KF. In a global localization problem, the initial state cannot be represented by a unimodal MVN as stated in Equation 3.16. Otherwise to trully encode the notion of a unknown initial belief, the covariance Σ_0 should be infinite.

Extensions and variants of the KF could be implemented to handle some of the addressed limitations, as a *Multi-Hypothesis Kalman Filter* (MHKF), where a mixture of Gaussian distributions is applied to cope with multimodal distributions. In the case of non-linear systems, a linearization of the state transition and measurement models can be performed as in the *Extended Kalman Filter* (EKF), where the linearization is obtained via a first-order Taylor series expansion. The Unscented Kalman Filter (UKF)[34] uses a stochastic linearization with a weighted statistical linear regression process. However, all these techniques are constrained to cases where the Gaussian-linear assumption is a suitable approximation, otherwise a more flexible posterior representation is needed.

Algorithm 2 Kalman Filter Pseudocode

```
procedure KalmanFilter($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)
    $\overline{\mu}_t = A_t \mu_{t-1} + B_t u_t$
    $\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R$
    $K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q)^{-1}$
    $\mu_t = \overline{\mu}_t + K_t (z_t - C_t \overline{\mu}_t)$
    $\Sigma_t = (I - K_t C_t) \overline{\Sigma}_t$
return $\mu_t, \Sigma_t$
```

3.1.3 Particle Filter

Particle filter (PF) [62] is called a nonparametric filter, what means that the probability distributions involved in the state estimation does not depend on a fixed set of parameters to be represented. Therefore the filter has more flexibility and the distributions are not limited to a fixed funtional form, as in the case of the Kalman Filter. This characteristic allows particle filters to solve the global localization and the kidnapped robot problems, that were remaining as open problems. The PF is a variation of a Sequential Monte Carlo Filter [5] and approximates the internal belief of the system by a set of N random samples $x_t^{[m]}$, or *particles*, drawn from the internal belief of the system $bel(x_t)$. A Monte Carlo estimator approximates the a probability distribution as:

$$p(x) = \frac{1}{N}\sum_{m=1}^{N} \delta_{x_t^m}(x_t), \tag{3.22}$$

where $\delta_{x_t^m}$ denotes the Dirac delta point mass function centered on x_t^m. If there are enough samples, this nonparametric approximation can represent a much broader and complex space of distributions. Strongly nonlinear systems and multimodal distributions are well handled by this type of filter. However, this flexibility comes with a high computational cost as the number of particles increases.

The key idea is that each particle $x_t^{[m]}$ represents a concrete instantiation of the state x_t and the posterior distribution $bel(x_t)$ is approximated by the set of particles $\chi_t := x_t^{[1]}, x_t^{[2]}, \ldots, x_t^{[M]}$. To the representation be accurate, each particle $x_t^{[m]}$ should have a probability to be drawn and included in χ_t proportional to the posterior probability $bel(x_t)$ or, in other words, proportional to how likely to be true the state instantiation $x_t^{[m]}$ is:

$$x_t^{[m]} \sim p(x_t | z_{1:t}, u_{1:t}). \tag{3.23}$$

Eq. 3.23 implies that the regions of the state space with a greater concentration of particles will have a higher likelihood of the true state to be located in that region. Although, sampling directly from any arbitrary function or probability distribution is not always possible and to overcome this problem a method called *Importance Sampling* (IS) is applied. The key idea of IS is to approximate a *target* distribution $f(x)$ by a set of weighted samples drawn from a second distribution $g(x)$, called *importance* distribution. This distribution is considered to be easy to sample from. Figure 3.1 illustrates this concept. Supose that we want to estimate the target distribution f represented in red by a set of samples (blue). However it is only possible to sample from a proposal distribution g (green), illustrated in 3.1(b). By weighting each sample it is possible to represent the target distribution by the samples from the proposal (3.1(c)). To understand the method first consider the expected value of an arbitrary function h(x) over the *target* distribution as given as:

$$E_{h(x)}(f(x)) = \int h(x) f(x) dx. \tag{3.24}$$

By assuming that f and g have the same support, i.e if $f(x) > 0 \rightarrow g(x) > 0$. The integral in 3.24 can be rewritten as, without any harm

$$E_{h(x)}(f(x)) = \int h(x) \frac{g(x) f(x)}{g(x)} dx. \tag{3.25}$$

As the *importance* or *proposal* distribution is choosen to be easily sampled from, it can be approximated by its Monte Carlo estimation (Eq. 3.22). Equation 3.25 can be then approximated as

$$\int h(x)\frac{f(x)}{g(x)}\frac{1}{N}\sum_{i=1}^{N}\delta_{x_t^m}(x_t)dx \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x^m)}{g(x^m)}h(x^m). \tag{3.26}$$

If we define $w^m = \frac{p(x^m)}{g(x^m)}$, it is possible to reformulate the approximation of the *target* distribution $f(x)$ as an equivalent Monte Carlo approximation from weighted samples from the *proposal* distribuition:

$$f(x) \approx \frac{1}{N}\sum_{i=1}^{N} w^m \delta_{x_t^m}(x_t). \tag{3.27}$$

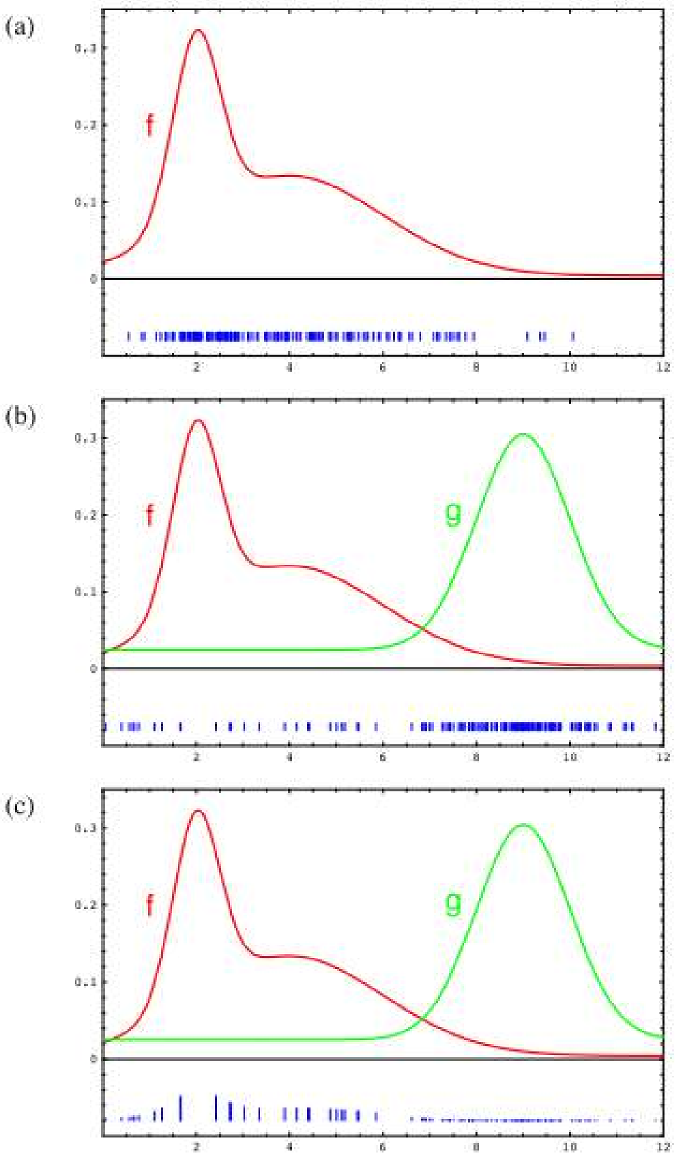


Figure 3.1: Importance sampling exemplification [63]

The proposal distribution must be carefully choosen to approximates the *target* distribution $bel(x_t)$. Due the recursive nature of the filter, each particle $x_t^{[m]}$ is generated based on the particle $x_{t-1}^{[m]} \in \chi_{t-1}$, which is assumed to be distributed according to $bel(x_{t-1})$. The proposal distribution is then defined as:

$$p(x_t|x_{t-1}, u_t)bel(x_{t-1}), \tag{3.28}$$

what involves the ability to sample from the *state transition probability.* This topic will be deeper discussed when the motion model for an underwater vehicle is presented in Section 3.3. To define the weight $w^{[m]}$ of each particle it is useful to consider the particle filter carries the whole sequence of states from which particle was generated $x_{0:t}^{[m]} = x_0^{[m]}, x_1^{[m]}, \dots x_t^{[m]}$. This is an artifact to help with the weight calculations and with this concept in mind it is possible to define the posterior over all states sequences:

$$\begin{aligned}
bel(x_{0:t}) =& p(x_{0:t}|z_{1:t}, u_{1:t}) \\
\overset{\text{Bayes}}{=}& \eta p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t})p(x_{0:t}|z_{1:t-1}, u_{1:t}) \\
\overset{\text{Markov}}{=}& \eta p(z_t|x_t)p(x_{0:t}|z_{1:t-1}, u_{1:t}) \\
=& \eta p(z_t|x_t)p(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t})p(x_{0:t-1}|z_{1:t-1}, u_{1:t}) \\
\overset{\text{Markov}}{=}& \eta p(z_t|x_t)p(x_t|x_{t-1}, u_t)p(x_{0:t-1}|z_{1:t-1}, u_{1:t-1}).
\end{aligned} \tag{3.29}$$

The weight of each particle can be then defined as:

$$\begin{aligned}
w_t^{[m]} &= \frac{target\,distribution}{proposal\,distribution} \\
&= \eta \frac{p(z_t|x_t)p(x_t|x_{t-1}, u_t)p(x_{0:t-1}|z_{1:t-1}, u_{1:t-1})}{p(x_t|x_{t-1}, u_t)p(x_{0:t-1}|z_{1:t-1}, u_{1:t-1})} \\
&= \eta p(z_t|x_t).
\end{aligned} \tag{3.30}$$

Particle weights or *importance factors* are responsible for incorporating the measurement z_t into the particle set. The weights describes how likely the measurement z_t is considered to happen under the state hypothesis $x_t^{[m]}$.

The particle filter algorithm, as represented in Algorithm 3 estimates the posterior $bel(x_t)$ from the previous belief, or in the particular representation, estimates χ_t based on the particles from χ_{t-1}. There is however a remaining step on PF algorithm, described in lines 7 to 9. The *Resampling* or *Importance Resampling* step draws samples with replacement from the posterior distribution with a likelihood proportional to their weights. Low weight particles will have a higher chance to be replaced by high weighted particles, which are more likely to be closer to the true state. IR does not change the number of particles, it just modifies how they are distributed, redistributing a weighted approximated density to an unweighted density. The particle set $\overline{\chi}_t$ that was originally distributed according to the prior distribution $\overline{bel}(x_t)$ is resample to be approximately distributed according to the posterior $bel(x_t) = \eta p(z_t|x_t^{[m]})\overline{bel}(x_t)$.

The resampling step reduces the covariance of the state estimation, however it induces a loss of divertisity in the particle set. Therefore, it should not be performed at every iteration step, with the risk of ending up with only one dominant particle. On the other side, if one resamples the particle set too seldom or not at all, the particles will start to diverge, and may end up in regions with low probability. The periodicity in which the resampling occurs must be well tuned. The most common criteria to analyse the degeneracy of the particles distribution is the *effective sample size* (or *effective number of particles*):

$$N_{eff} = \frac{1}{\sum_{m=1}^{N}(w_t^{[m]})^2}. \tag{3.31}$$

Once N_{eff} drops below a threshold value e.g. $N_{eff} < \frac{N}{2}$, the particles the resampling step takes place, otherwise the weights are updated according to [26]:

$$w_t^{[m]} = p(z_t|x_t^{[m]})w_{t-1}^{[m]}. \tag{3.32}$$

Another side effect of the resampling step is called *particle deprivation* and occurs due to the randomness nature of the filter. A series of random particle replacements can wipe all the particles near the true state and the filter would struggle to recover from a wrong convergence. This drawback can be avoided introducing a small set of randomly generated particles as implemented in [45] within the localization system of an AUV. This method deteriorates the state estimation as introduces particles that are not necessarly near the true state and should not be performed at every step. A practical implementation is to spread some particles if the maximum value of the particles weights is bellow a threshold.

3.2 State Vector

To localize an underwater vehicle in a 3D complex environment it is necessary 6*Degrees of freedom* (DOF) to describe its pose. Three for the position of the system and three for its orientation with respect to an inertial reference frame $\Sigma_i(O_i - xyz)$. In low speed underwater systems, a fixed-earth frame can be considered as fixed inertial reference frame. The body-fixed reference frame $\Sigma_b(O_b - x_b y_b z_b)$ of an underwater vehicle has its origin O_b usually placed on the center of gravity of the vehicle when this point is in the principal plane of symmetry or at any arbitrary

Algorithm 3 Particle Filter Pseudocode

```
ParticleFilter($\chi_{t-1}, \Sigma_t, u_t, z_t$)
$\overline{\chi}_t = \chi_t = \emptyset$
for $m = 1$ to $M$ do
    sample $x_t^{[m]} \sim p(x_t|u_t, x_{t-1}^{[m]})$
    $w_t^{[m]} = p(z_t|x_t^{[m]})w_{t-1}^{[m]}$
    $\overline{\chi}_t = \overline{\chi}_t + \langle x^{[m]}, w^{[m]} \rangle$
if $N_{eff} < M/2$ then
    for $m = 1$ to $M$ do
        draw $i$ with probability $\propto w_t^{[i]}$
        add $x_t^{[i]}$ to $\chi_t$
return $\chi_t$
```

point for better convenience. The orientation of the body coordinate frame in marine vehicles is aligned with the principal axes of inertia [21], as ilustrated in Figure 3.2.:

- x_b -longitudinal axis (pointing forward)
- y_b - transversal axis directed to right-hand facing forward, or starboard
- z_b - normal axis (directed from top to bottom)

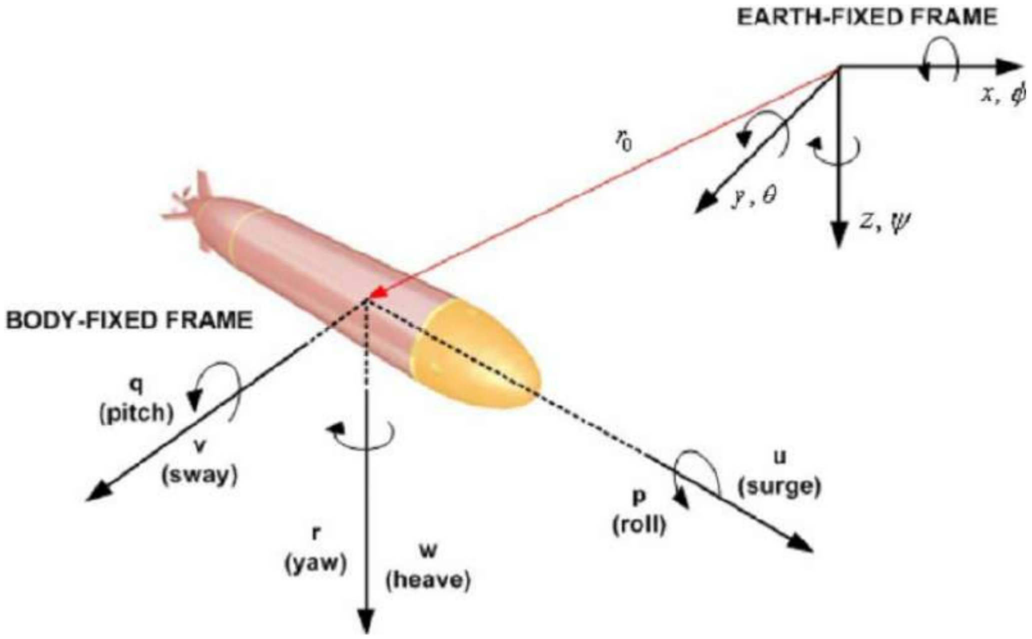


Figure 3.2: Inertial and body-fixed coordinate frames

Lets define $\eta_1 \in \mathbb{R}^3$ as the vector of the body position in the inertial reference frame

$$\eta_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{3.33}$$

The orientation is defined by the vector $\eta_2 \in \mathbb{R}^3$ that represent Euler-angles for roll (ϕ), pitch (θ) and yaw (ψ), also relative to the fixed reference frame.

$$\eta_2 = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}. \tag{3.34}$$

The orientation of the sytem can be represented also by a rotation matrix $R_{ab} \in SO(3)$, that rotates the coordinate frame Σ_b to Σ_a. $SO(3)$ is the *special orthogonal group of order 3*:

$$SO(3) = \{R \in \mathbb{R}^{3\times3} : RR^T = I_{3\times3}, det(R) = 1\}, \tag{3.35}$$

and $I_{3\times3}$ denotes the *Identity* matrix of order 3. The orientation of the vehicle in the inertial frame can be expressed by the composition of elementary rotations, computed via premultiplication, as $R_{ib} = R_z(\psi)R_y(\theta)R_x(\phi)$ where R_l is a rotation matrix around the *l-axis*.

$$R_{ib} = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}, \tag{3.36}$$

where $s\beta$ and $c\beta$ stands for $sin(\beta)$ and $cos(\beta)$, respectively. In conclusion, the state vector that fully describes the pose of the vehicle relative to the inertial reference frame is

$$x_t = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = [x, y, z, \phi, \theta, \psi]^T. \tag{3.37}$$

Although as discussed in 3.1.3, the convergence of the Monte Carlo estimation depends on the number of samples. Therefore, it would require a high number of particles to achieve a good approximation of the posterior distribution over a six dimensional state variable.

The proposed localization method is not specific to a given particular underwater vehicle or system, however there are some assumptions that can be made without compromising the validity or how generic the solution is. Most of the underwater vehicle are passively stable in roll and pitch axis and, if not, it can be considered that a good absolute measurement can be achieved with the usage of a IMU and a DR system as discussed in Section 2.1. For this reasons, the state estimation can directly use the values for the roll and pitch axis, with a fixed error distribution. The measurements from a calibrated pressure sensor returns an estimation of the vehicle's depth (position in the $z-axis$) and can have zero mean Gaussian error associated with variance σ_z^2. On the other side, a compass measurement can be affected by ferromagnetic structures, specially in man-made environment as hydroelectric plants, and therefore can be biased. Finally, as the measurement model, discussed in Section 3.4, is not continuous in the x and y axis and these dimensions still need to be sampled.

In summary, depth, pitch and roll are considered as instrumented with strapdown high update sensors and provide a direct measurement of z, ϕ, ψ. There is no direct sensor for the remaining dimensions and therefore the PF has to sample these degrees of freedom.

$$x_t = (\overbrace{x, y, \theta}^{\text{sampled}}, \underbrace{z, \phi, \psi}_{directly\ measured})^T. \tag{3.38}$$

3.3 Motion Model

As introduced in Section 3.1, the state transition probability $p(x_t|x_{t-1}, u_t)$ plays a fundamental role in the prediction step of any practical Bayesian Filtering implementation. Aiming to maintain a solution as generic as possible, the *motion model* of the vehicle is assumed as a *kinematic model*. More complex dynamic models are hardly available and are highly dependent on specific parameters that strongly differ for each different vehicle. Design aspects such as position and orientation of the vehicle's thursters, weight distribution, buoyancy and drag effects may drastically affect how the vehicle behaves and therefore its model. The kinematic model describes, basically, the motion of the robot, without considering the forces that caused it. The vehicle is represented as a rigid body, i.e the distance between any two given points of the vehicle is considered constant. As stated in Section 3.2, it is assumed the presence of a DR system and the localization loop has direct access to the linear and angular velocities represented on the vehicle coordinate frame as:

$$\nu = \begin{bmatrix} \nu_1 \\ \nu_2 \end{bmatrix} = [u, v, w, p, q, r]^T, Section \tag{3.39}$$

where $\nu_1 = [u, v, w]^T$ is the linear velocities vector of the origin of the body-fixed frame Σ_b related to surge, sway and heave, respectively, and $\nu_2 = [p, q, r]^T$ is the angular velocites vector related to the roll, pitch and yaw.

As the proposed localization method implements a particle filter, there is no need to compute the entire state transition probability distribution for arbitrary x_t, u_t and x_{t-1}. Instead, it suffices to be able to sample from this distribution as discussed in Section 3.1.3. The abitily to sample from the motion model means that given u_t and x_{t-1}, one must be able to generate a random instantiation of x_t drawn according to $p(x_t|u_t, x_{t-1})$. Each particle state is thus updated considering the displacement travelled by the vehicle during the period Δ_t between two consecutives time instants, assuming a constant velocity within the interval and Gaussian error. Firstly, let us define the deterministic kinematic model of an underwater vehicle. Thorpe and Whyte [61] summarized a generic 6-DOF model for an underwater vehicle. As the linear and angular velocities are expressed in the body-fixed coordinate frame, it is necessary to perform a coordinate frame transformation to derive the state. The linear velocities of the vehicle can be expressed in the inertial coordinate frame as

$$\dot{\eta}_1 = R_{ib} \cdot \nu_1, \tag{3.40}$$

where the vector $\dot{\eta}_1 \in \mathbb{R}^3$ is the time derivative of the position vector η_1 expressed in the inertial frame. The vector $\dot{\eta}_2 \in \mathbb{R}^3$, expressed in the inertial frame, is the time derivative of the orientation and is related to the body-fixed angular velocity by a proper Jacobian matrix

$$\dot{\eta}_2 = J_{k,o}(\eta_2)^{-1}\nu_2, \tag{3.41}$$

where

$$J_{k,o}(\eta_2) = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix}. \tag{3.42}$$

It is worth noting that the Jacobian matrix is not inversible for every pitch angle value θ. There are singularities for $\theta = \pm(2l+1)\frac{\pi}{2}$ rad, with $l \in \mathbb{N}$, i.e. when the vehicle is in full vertical orientation, what is not common for this class of vehicles. Otherwise, a quaternion representation should be preferred. The transformation between the velocities expressed in body-fixed frame to inertial reference frame can be then expressed as

$$\dot{x}_t = \begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} R_{ib} & O_{3\times 3} \\ O_{3\times 3} & J_{k,o}^{-1} \end{bmatrix}}_{J} \nu. \tag{3.43}$$

Considering the velocities constant within the time interval Δ_t, the evolution of the pose of the vehicle can be expressed as:

$$x_t = x_{t-1} + J(x_{t-1})\nu\Delta_t. \tag{3.44}$$

Once Equation 3.44 is defined, the probabilistic model can be then derived. The velocities represent the control data or input u_t, as they carry the information about the change of state, or position, perfomed by the robot. Considering that the actual velocities differ from the inputs ν, a zero mean Gaussian additive noise, with covariance Σ_ν, is associated to the velocities readings:

then define the controle signal $u_t = \hat{\nu}$, where$\hat{\nu}$ is given by

$$\hat{\nu} = \nu + \mathcal{N}(0, \Sigma_\nu), \tag{3.45}$$

$$\Sigma_\nu = \begin{bmatrix} \sigma_u^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_v^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_w^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_p^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_q^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_r^2 \end{bmatrix}. \tag{3.46}$$

The process noise also accounts for the uncertainty introduced by the simplifications when considering a kinematic model instead of a dynamic one. If a minimum update rate, i.e an small Δ_t, is satisfied, this model can estimate the motion of an underwater vehicle with good accuracy and the divergences to more complex models are attenuated.

Therefore, each particle is updated with a random velocity value ν_t^*, drawn from the probability distribution $\mathcal{N}(\nu*, \nu, \Sigma_{nu})$. The particle $x_t^{[m]}$ is generated by a disturbed velocity vector integrated over the previous state instantiation $x_{t-1}^{[m]}$. This randomness in prediction step describes the uncertainties in the process and also helps the filter to explore new regions of the state-space after convergence, as the particles are spread.

3.4 Measurement Model

The additive nature of the noise present in the motion process also tends to diverge the estimation of the true state and, as discussed before in section 3.1, a measurement must be introduced to correct the internal belief of the system. The measurement model, which implements the *measurement probability distribution* (Eq. 3.3), focuses on describing the intrinsec uncertainties of the measurement generation process. As the main idea of probabilistic robotics, an accurate sensor model is often hard, or even impossible, to achieve and not always desirable. The modelling of a perfect measurement model demands a high time-wise cost and can rely on unobservable variables. Thus, the conditional probability distribution $p(z_t|x_t)$ enconpasses the uncertainties and returns an evaluation of the probability of a measurement z_t to happen given the state x_t.

As discussed in section 2.3.2, the mechanically scanned sonar Tritech Micron is becoming a popular choice in the development of small underwater vehicle such as ROVs and AUVs, due its cheap price and reduced weight and size. However, as ranging sensor theory is already mature, imaging sonar are applied mainly as really noisy range finder sensor. In addition, authors consider imaging sonar as 2D scan sensor, as a result of the vertical ambiguty introduced by the wide vertical beamwidth. This section will formalize the ranging model and present the proposed imaging or intensity sensor model.

3.4.1 Range Finder Model

The range finder model approximates a sonar beam (or a laser beam) as an ideal unidimensional ray and measure the range to the first object along the direction of travel. The noise in this range measurement, i.e the error between a reference or true range z_t^* and the measured range z_t, is usually modeled by a narrow Gaussian with mean z_t and standard deviation σ_{hit}[63]. This error arises from the limited resolution of range sensors, environmental aspects as atmosferic effect for airborne systems or water pressure and temperature for underwater sensors. On LIDAR systems, that Gaussian is fairly narrow, while on sonars its profile is much wider.

$$p_{hit}(z_t|x_t, m) = \begin{cases} \eta\mathcal{N}(z_t, z_t^*, \sigma_{hit}^2), & \text{if } 0 \leq z_t < z_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3.47}$$

where η is a normalizer variable and the values measured by the range sensor are limited to its maximum range z_{max}. The computation of the range z_t^* is usually made by a ray-casting process, where the distance to first object intersected by a ray is assumed as the reference measurement. Even though this model can be robust enough to encompass the uncertanties and noise provenient from the sensor and was sucessfully implemented within underwater vehicle's localization systems, there is a limitation on the represetantion of the environment and only 2.5D (elevation maps) can be represented. While the ray-casting method well represents the extremely narrow beam of a laser sensor, the wide vertical opening of imaging sonars cannot be represented by a single unidimensional ray and the simulated rays can be only casted horizontally. Another important characteristic of considering an imaging sonar as a range sensor is that all the values regarding the intensity of the echoes are discarded as the measurement model cannot provide a reference to compare them.

3.4.2 Imaging sonar sensor model

To represent the beam shape of an imaging sonar and be able to cast a reference beam inside an environment it is necessary to introduce some sonar simulation theory.

Sonar Simulation

The propagation of sound in water, with a sound source located at (x_s, y_s, z_s) and modelled as punctual with a Dirac delta (δ) spatial distribution function, can be described by the acoustic version of the wave equation [16]

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = -\delta(x - x_s, y - y_s, z - z_s)s(t), \tag{3.48}$$

which considers a second order differential equation for the acoustic pressure p, as a function of the of space (x, y, z) and time (t) related to the point source. Equation 3.48 assumes water density constant and the speed of sound c known or measured. In addition, if one assumes the case of a infinite water mass around the acoustic signal source, the acoustic pressure can be expressed as [10]:

$$p(x, y, z, t) = \frac{s(t - \frac{r}{c})}{4\pi r}, \tag{3.49}$$

where r is the range from the point source:

$$r = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2} \tag{3.50}$$

The implementation of a sonar simulation following this model can achieved with a ray cast implementation as seen in [4, 18], where the volume insonified by the sonar beam is discretized by multiple rays, what comes with a high computational cost. To improve processing speed, *Guériot and Sintes* [23] used a tube tracing approach, in which the wave front is divided in subregions. Rays are used as support for a volume description of the insonified area, and the tubes are built on the top of neighboring rays using four of them as borders. This approach reduces the number of rays casted and handles better the intersection or footprint of the incoming sound with the objects facets.

A frequency domain analysis of the model presented in Eq. 3.48 was performed by *Coiras and Groen* [10], where a simulator for a side-scan sonar was implemented and in [58], the proposed method was adapted for a forward looking imaging sonar. The simulator considers the surface of simulated objects are discretized in facets or k surface patches and after several simplifications a intensity model for the contribution of a point r can be defined as:

$$I(r) = K\Theta(r)R(r)\hat{n} \cdot \hat{r} \tag{3.51}$$

where Θ represents the intensity of the illuminating sound wave $s(t)$ at point r and group all the beam-pattern properties as intensity variations caused by the sensor's beam-profile, spherical spreading loss, TVG and other corrections. R is the reflectivity of the of the insonified region, K is a normalization constant and $\hat{r}$, $\hat{n}$ are the vectors from the origin of the sensor to the surface point and its normal vector respectively. This states that the scattering model corresponds to Lambertian illumination model for diffuse reflectance, and the perceived brightness of a surface is independent of the direction of observation and depends only on the angle between the incidence beam and the surface patch normal.

Eq. 3.51 allows the simulation of a sonar beam, composed by an array of intensities values outputed, defined as

$$I_s = [i_s^1, i_s^2, \dots, i_s^N] \tag{3.52}$$

where i^n is the intensity value on the n-th element of the array and N is its size. Given the measured intensity values I_r by the sonar, an error function between the simulated and measures intenisties is defined as gaussian distribution centered in the simulated intensity i_s^n with variance σ_{int}^2

$$p(i_r^n|i_s^n) = \mathcal{N}(i_r^n, i_s^n, \sigma_{int}^2) = \frac{1}{\sqrt{2\pi\sigma_{int}^2}} e^{-\frac{(i_s^n - i_r^n)^2}{2\sigma_{int}^2}} \tag{3.53}$$

The measurement probability function $p(z_t|x_t)$ can be described by the contribution of every intensity received by the sonar and should reflect how similar the two arrays are

$$p(z_t|x_t) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{N}(i_r^n, i_s^n, \sigma_{int}^2) \tag{3.54}$$

The weight of the particle is then updated following $w_t^{[m]} = p(z_t|x_t^{[m]})w_{t-1}^{[m]}$ as in Eq. 3.32. The representation of the likelihood function as a Mixture of Gaussians in oposition to a pure Gaussian likelihood, was preferred to avoid a high sensitivity to noise. As a single near zero value would be dominant in the computation of the product of the single likelihood function presented in Eq. 3.53.

Chapter 4

Implementation and Results

4.1 Software Overview

The filter was implemented using the *Robotics Construction Kit*(ROCK)[33][1]. ROCK is a software framework specifically developed for the easy development and deploy of robotic systems. As in the last years the scale and scope of robotics had drastically grown, many frameworks came up to fulfill the need of this demanding area, as rewrite everything from scratch each time a new system is deployed has no logical sense. The two most popular available frameworks are ROS (Robot Operating System) [51] and the Orocos RTT (Real Time Toolkit) [60]. ROCK is component based, what means that there is a separation between the communication layer and specific functionalities. These contrasts with the common approach of developing components or nodes, whose behavior is very often dependent on many of internal states, which are therefore hard to assess externally. As the framework is based on the Orocos RTT, real-time application can be easily implemented and this fact is a key advantage in modern robotic systems.

All core funcionality code is written in C++ and Ruby scripts are used to start, connect and monitor components. As mentioned, there is a separation between the functionality layer and communication layer. The funcionalities are encapsulated in libraries and remain independent of the framework and can be reused or maintained outside ROCK environment. The components, on the other side, are responsible for integrating the functionality of the library, exposing its functionality to the system. Figure 4.1 illustrates how the library was constructed and its main funcionalities.

The particle filter class implements the generic functionalities discussed in Section 3.1.3 and the pose particle class stores a $x - y$ position 2D vector and a double for the yaw orientation value and weight. As explained in section 3.4, the generation of a meaningful reference measurement value z_t^* plays a fundamental role

[1] http://rock-robotics.org

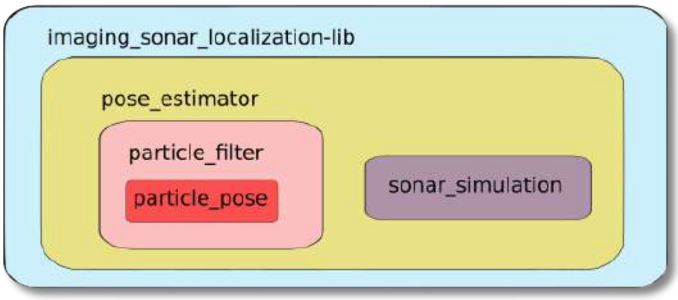


Figure 4.1: Imaging sonar localization and its main functionalities.

in the weight evaluation of each particle. The sonar simulation class is, then, responsible for generating the reference measurement and its implementation will be discussed further in Section 4.1.1. The Pose Estimator class implements the specific probabilistic distribution models discussed through Sections 3.3 and 3.4.2.

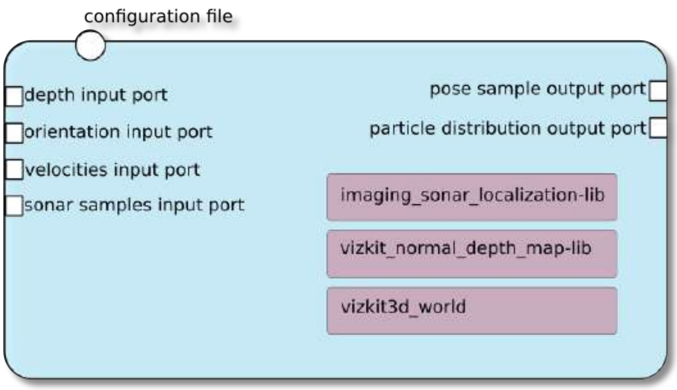


Figure 4.2: Imaging sonar localization component.

The *imaging sonar localization* component (Figure 4.2) is responsible for initialization, configuration, communication and computation of the filter. As well manage all ports and align the incoming data for a correct processing.

The *vizkit3d-world* is responsible for the representation of the underwater scene and its components, such as robots or underwater vehicles. The elements present in the environment are defined by means of SDF[2] (Simulation Description Format) files, an XML format used to describe simulated models and environments. As illustrated in Figure 4.3, this representation allows the simulation of multiple environments types and the representation of complex structures, such as strutuctural components, underwater machinery, etc.

[2] http://sdformat.org

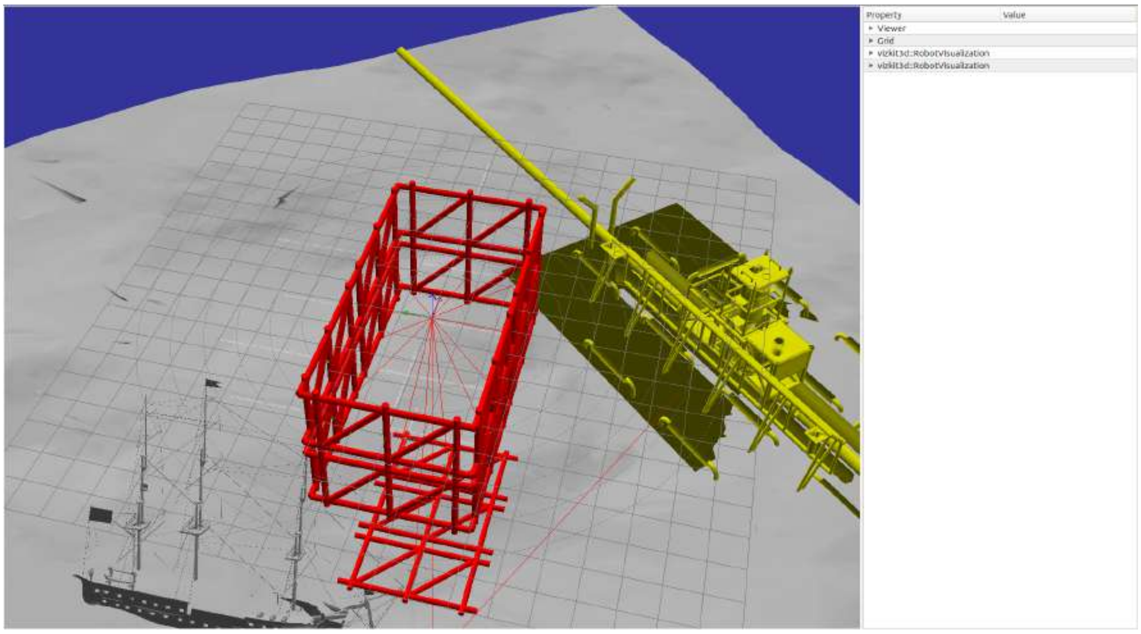


Figure 4.3: Representation of different complex underwater environments using SDF files.

4.1.1 Sonar Simulation Implementation

The sonar simulation was based on the method proposed by *Cerqueira et al.* [8], where a simulator of both FLS and MSIS sonars that relies on the parallel processing power of modern Graphical Processing Units (GPU) to speed up 3D graphics processing. The simulator is written in C++ as a ROCK package and the framework's graphical engines, based on the OpenSceneGraph[3] library, is responsible of the 3D rendering.

The sonar sensor is represented as a camera of the 3D rendering process, with same horizontal and vertical field of view (FOV-X, FOV-Y) specified by the type of sonar in the configuration file of its component.

Sonar data are simulated by a shader process based on OpenGL Shading Language (GLSL)[4] which allows to handle the rendering pipeline executed on the GPU. Its output is a 3-channel matrix that contains the intensity, range and bearing for each part of the 3D underwater scene that is visible from the viewpoint of the sonar sensor.

The first channel is the bearing representation in the camera, or angle distortion channel, and it is reponsible for addressing the regions of the rendered image to its respective beams. As all pixels in a column of the angle distortion channel image have the same angle value, then each column from the other two channels can be subdivided in separate beams or subimages. The angle distortion matrix has 0.0

[3] http://www.openscenegraph.org/

[4] http://www.opengl.org/documentation/glsl/

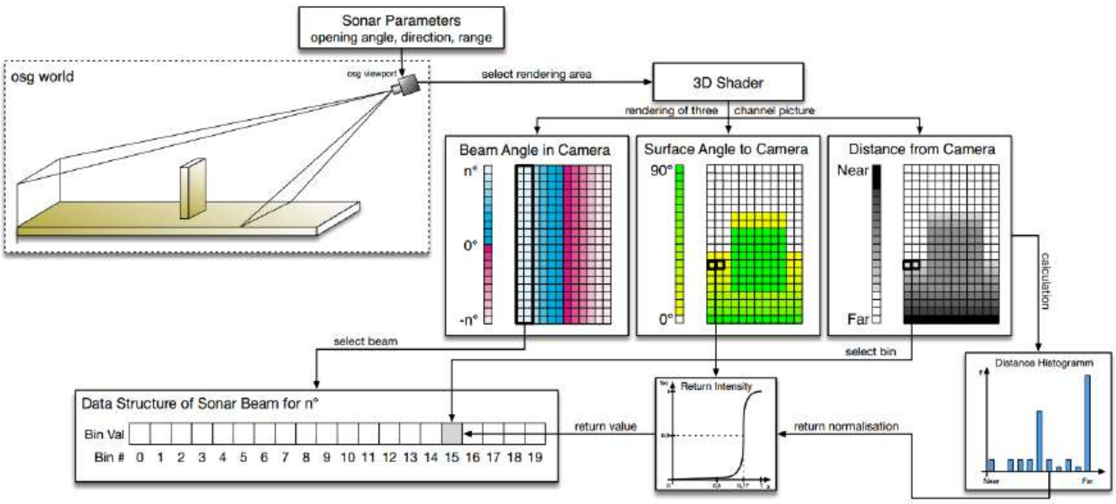


Figure 4.4: A graphical representation of the individual steps to get from the OpenSceneGraph scene to a sonar beam data structure. ([8]).

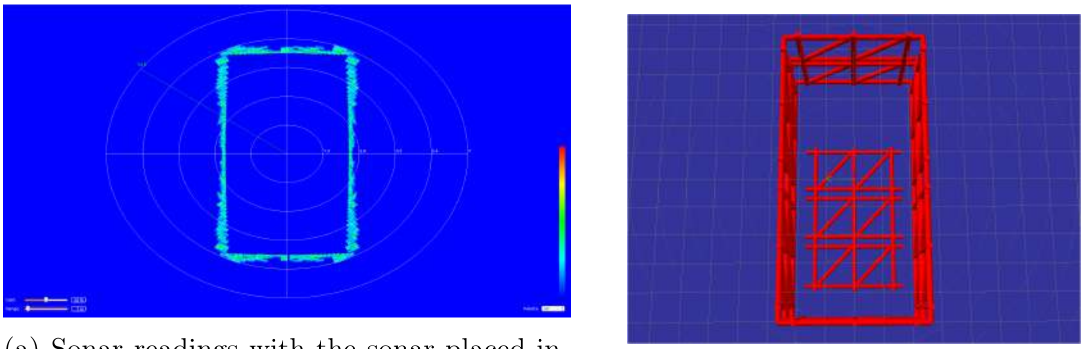
value in the center column, and 1.0 value in both border columns which represents FOV-X half value. In the case of a single beam (MSIS) sonar, all the columns are mapped to the unique beam and the head position of the sensor is automatically rotated to change the direction of the horizontal FOV.

The intensity channel simulates the echo refletion energy based on the dot product between the incident vector coming from the sonar ray and the surface normal, what is aligned with the Lambertian ilumination model presented in Eq.3.51. Finally, the third channel stores the depth or range data of the rendered area inside the FOV of the sonar. Each pixel is a depth measurement, i.e. the 3D Euclidean distance between the camera focal point and the object's surface point. For depth data, the minimum value portrays a close object while the maximum value represents a far object, limited by sonar max range. The convertion from the 3-channel matrix subimages to sonar data, represented by the array of bin intensities, is made using the depth and intensity values (with its respective columns addressed by the angle distortion matrix). Each bin intensity value is calculated based on the accumulated intensity contribution from every individual pixel intensity value that sits on a particular bin sector, according to the range value from the depth channel matrix. Therefore, the accumulated intensity of each bin I_{bin} is calculated as follows:

$$I_{bin} = \sum_{x=1}^{n} \frac{1}{n} sig(i(x)), \tag{4.1}$$

where n is the number of pixels that have the same depth value, i.e. the same distance from the sensor, $i(.)$ is the intensity value of the pixel and $sig(.)$ is a sigmoid function. In order to count how many pixels contribute to a given bin, a depth histogram is calculated and the bin number is proportional to the real distance from the sensor. In other words, the initial bins represent the closest distances and the last one represents the maximum specified range. Figure 4.4 illustrates the selection of a beam on the angle distortion matrix and the contribution of two pixels to the respective bin.

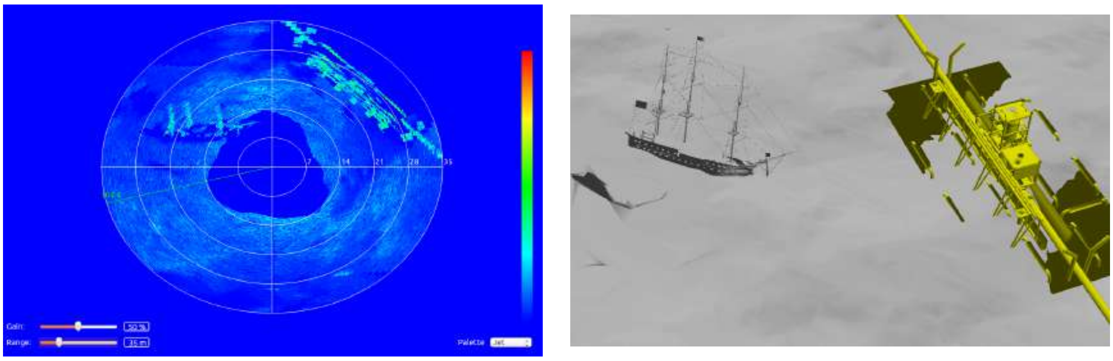
The simulator prioritizes efficiency and makes some simplifications, such as not specifying material properties to allow different refractions values depending on material type as in [15], and discard effects like multipath. However it is considerably faster, as for 150 sampling frames, the proposed method produced one multibeam sonar data every 121.44 milliseconds and one single beam sonar data every 8.5 milliseconds, if compared with the sampling rates achieved by the similar simulator presented in [15] (1 second) and [58] (2.5 minutes). The output of the sonar simulation can be visualized in Figure 4.5a, where the sonar sensor is placed inside a metal structure with closed form (Fig. 4.5b) . Figure 4.6b represents the sonar readings in a more complex structure such as a oil and gas underwater manifold and a sunken ship (Fig. 4.6a.



(a) Sonar readings with the sonar placed inside a metal framerwork structure

(b) Closed red metal framework map

Figure 4.5: Sonar Simulation in an structured environment.



(a) Sonar readings with the sonar placed between the two structures.

(b) Complex structures such as oil and gas manifold and sunken ship

Figure 4.6: Sonar Simulation in a unstructured environment.

4.1.2 Particle weight computation

In order to compute the weight of each particle $w^{[m]_t} \in \overline{\chi}$, its full pose must be constructed at the correction step. Each particle samples the dimensions $[x, y, \psi]$, which must be merged with the incoming data from the pressure sensor and the inclination from the last reading before the incoming sonar data. The sonar data structure is represented by a timestamp, representing the time that sample arrived, an angle value and an array of intensities, representing the bins. After the computation of each particle pose, the filter must be able to query a single simulated sonar beam for every particle. The 3D shader rendering process needed to be isolated and to cope with this requirement and all the vehicle visualization tools and vehicle physics simulation present on the Gazebo[5] framework were separated and only the sonar data was computed. For each simulated beam, the filter assigns a weight for the respecticve particle as defined in 3.54.

4.2 Simulations Results

The map used is a representation of a $10 \times 5m$ underwater structure, constructed with metal beams arranged in a trellis pattern. The map is defined in a XMl file and is illustrated in Figure 4.7. This environment represents the classical test scenario as it has the shape of a pool or structured environment with vertical walls. This scenario is also present in real application as the he Stoplog well (where the Stoplog are inserted to stop the water flow) in the context of the ROSA project, and is also explored by [52–55]. The competition SAUCE-E *Student Autonomous Underwater Challenge - Europe*[6] made it the stardard target, introducing many of it challenges inside a pool or in a Harbor. However, it is worth noting that the trellis structure introduces complexity in the map if compared to a concrete wall. A ray casting based algorithm, for example, would struggle with misreadings between the metal beams.

4.2.1 Tracking

The tracking problem consists in estimating the true state, given a good initial guess, and track its evolution during the trajectory performed by the vehicle. To evaluate this problem, the sonar sensor origin was considered at the vehicle or body coordinate frame. The parameters used in the simulations were chosen to optimize the perform of the algorithm and are listed in Table 4.1, where $bel(x_0)$ is the mean of the initial belief of the system, represented by a MVN with variances $\sigma_{0_{pos}}$ for the x

[5] http://gazebosim.org/

[6] http://sauc-europe.org

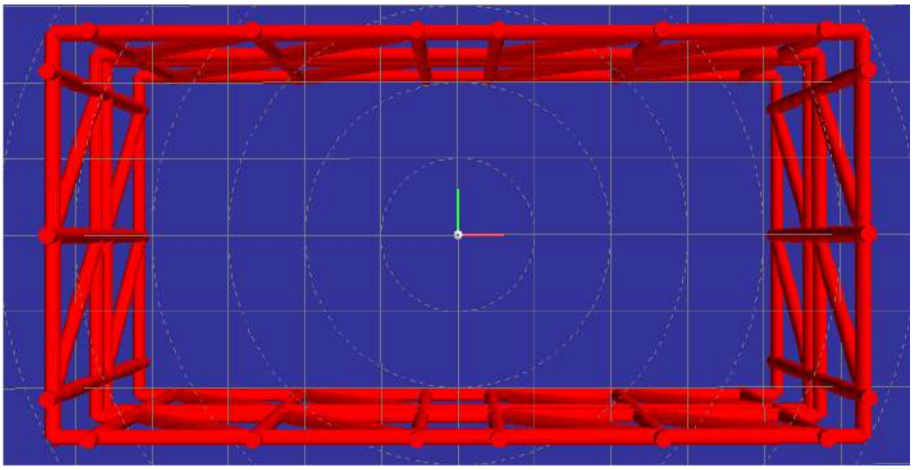


Figure 4.7: Test scenario of structured environment.

and y dimensions and $\sigma_{0_{ori}}$ for the yaw angle. σ_u, σ_v, σ_r are the significant variances in the velocities input (Eq. 3.45) and σ_{int}, σ_{sp_u}, σ_{sp_v}, σ_{sp_r} are the measurement model variance (Eq. 3.54) and the spread function variances for the linear and angular velocities. A further discussion in the impact of these parameters will be made in the final considerations at the end of this section.

Table 4.1: Parameters tracking problem.

$bel(x_0)$	x_0	$\sigma_{0_{pos}}$	$\sigma_{0_{ori}}$	σ_u	σ_v	σ_r	σ_{int}	σ_{sp_u}	σ_{sp_v}	σ_{sp_r}
$[-1, .25]$	$[-.5, .5]$	$[.7, .7]$m	.3rad	10%	10%	10%	.1	.2	.2	.02

To illustrate the evolution of the filter, lets consider the particles that were first spread around the position $[x_0, y_0] = [-1, 0.25]^T$ with variance equal 0.7 in both directions. With the arrival of the first sonar samples, the particles were resample to the distribution represented in the Figure 4.8. The yellow particles are defined as high weight particles and the red particles as low weight ones. This fact is due the readings provenient from the sonar that is represented by the coordinate frame and the surving particles were the ones within the same distance to the wall in the direction of the $x - axis$ (represented in red) and bearing close to zero. The ambiguity in the y position is because until this step, the sonar did not provide any information that allowed the filter to better distinguish the likelyhood of any particle lying in that line.

Figure 4.9 shows the results of the filter with different number of particles following a pre-defined trajectory, represented by the full green line. The trajectory was in coplanar with the xy plane to make the visualization easier and as while testing all the dimensions sampled by the filter. From the results analysis it is possible to see that the filter converges for each of the three simulations. Although, as the number of particles decreases, the uncertainty introduced by the velocity input noise has a higher impact in the filter estimation. There is no enough particles to keep tracking

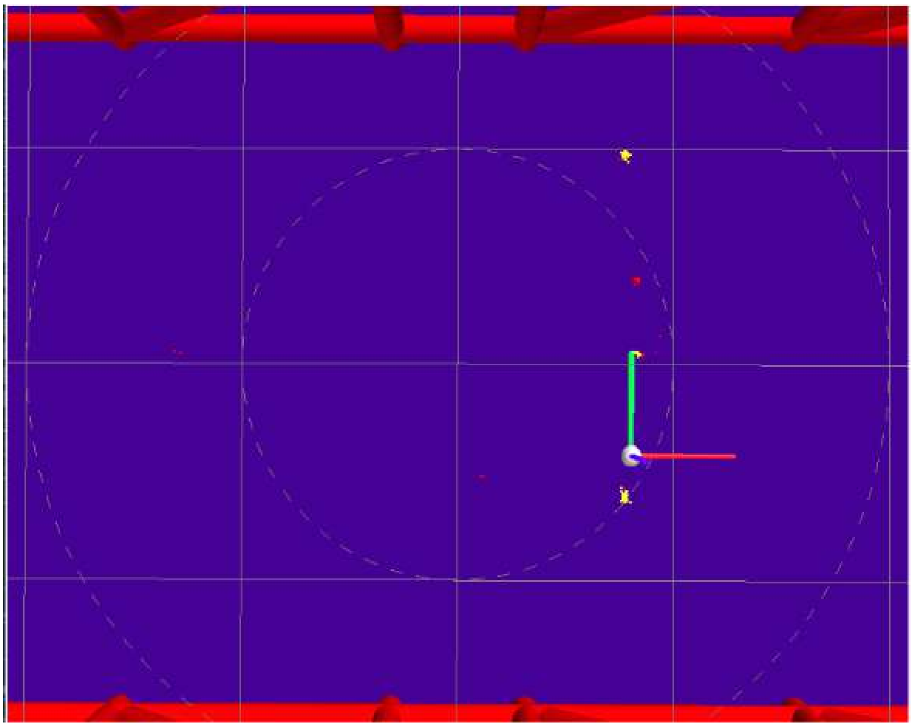


Figure 4.8: Test scenario of structured environment.

the true state and the filter starts to diverge. The filter has to spread the particles to push particles in a region of convergence, this effect is obvious in the simulation with 10 particles and it is illustrated in Figure 4.10. This effect illustrates also the capability of the filter to recover from a kidnapped robot problem. As the number of particles grows, the initial convergence speed is also increased, as the chances of particles to be initialized near the true state grows. However, for a tracking problem the difference between the number of particles above a certain level is not worth the computational effort. The 50 and 100 particles simulations have similiar results in the particular scenario.

4.2.2 Global Localization and kidnapped robot problem

The global localization problem introduces more complexity in the state estimation, because the the robot can be initialized in any region of the space state, what naturally demands a bigger number of particles. The parameters used in this simulations are the same in the tracking simulation, only increasing the initial spread to cover the entire map.

Table 4.2: Parameters global localization problem.

$bel(x_0)$	x_0	$\sigma_{0_{pos}}$	$\sigma_{0_{ori}}$	σ_u	σ_v	σ_r	σ_{int}	σ_{sp_u}	σ_{sp_v}	σ_{sp_r}
$[-1, 0.25]$	$[0, 0]$	$[3, 1]$m	1rad	10%	10%	10%	0.1	0.2	0.2	0.02

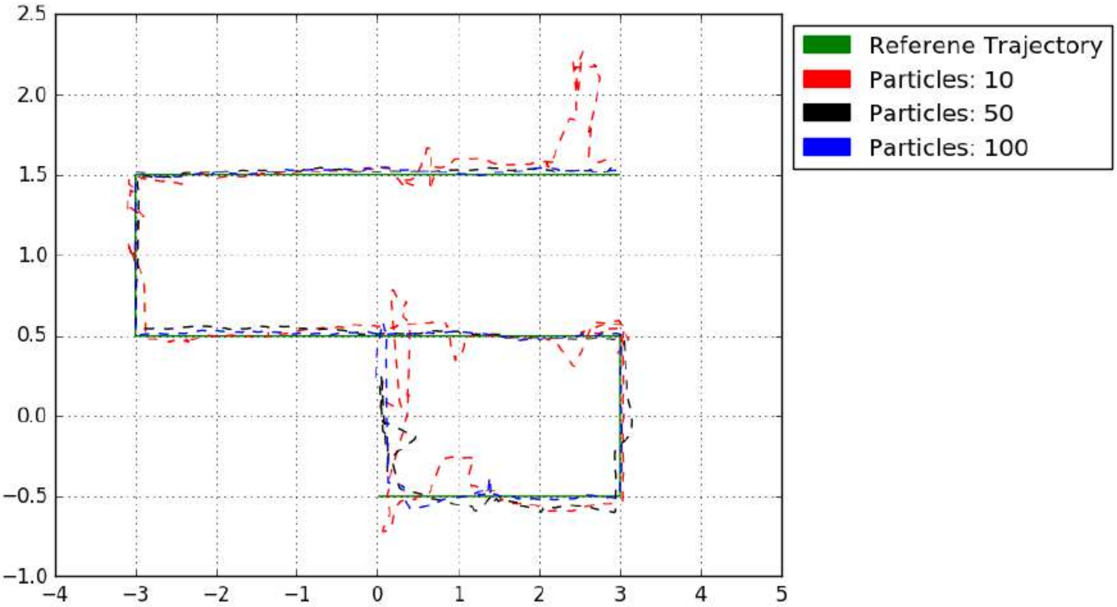
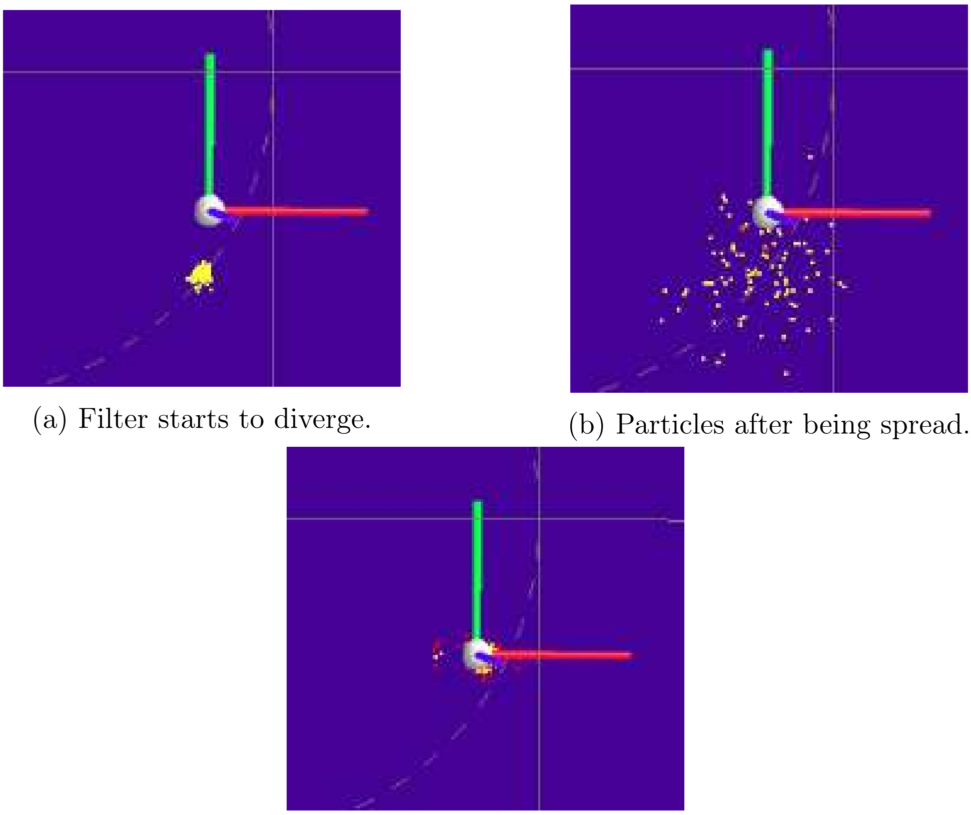


Figure 4.9: Tracking results for 10, 50 and 100 particles (v=0.1m/s).



(a) Filter starts to diverge.

(b) Particles after being spread.

(c) Reconvergence after the particles being spread

Figure 4.10: Illustration of the spread function avoiding the filter to diverge.

In this scenario the initialization of the filter with 10 particles was not enough to achieve a convergence to the true state, as can be seen in the Figure 4.12. With 100 particles, the algorithm sucessfully converged with incoming of the first sonar sam-

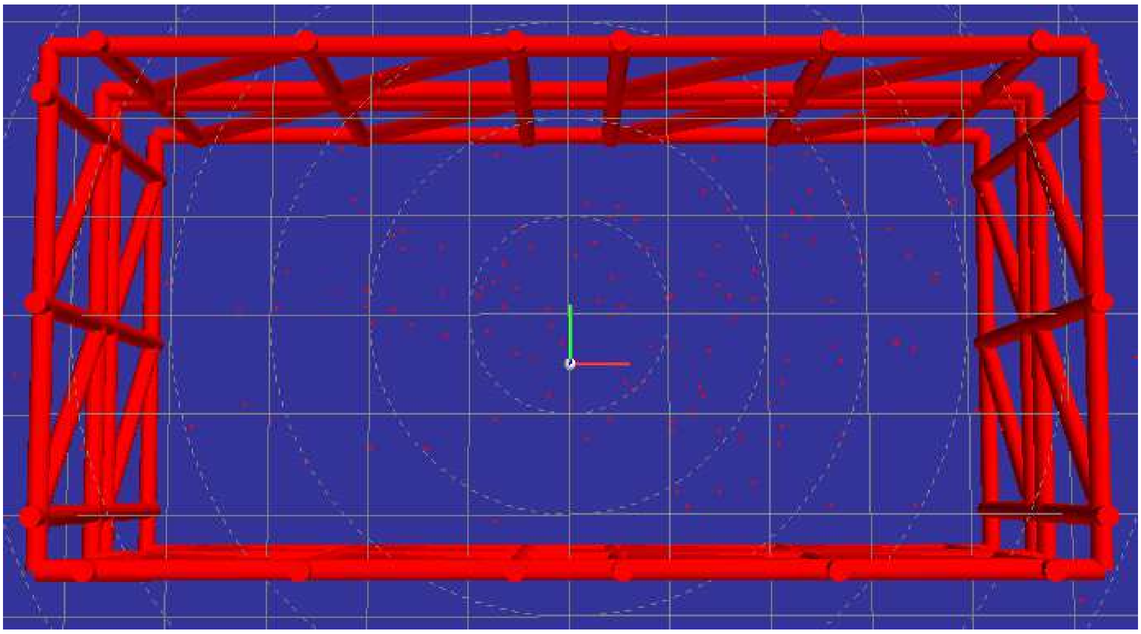


Figure 4.11: Particles spread in the entire map.

ples, as the particles were well spread over the state space. The case of 50 particles is a good example of the kidnapped robot problem, as the initial distribution did not covered the region of the true initial state and all the particles were associated with low weights. As result, the particles were bouncing around the map and at each iteration being more spread due the low maximum weight of the distribution. After a few steps some particles landed near the true state and the convergence was sucessfull. But it is needed to highlight the randomness of this process, what does not ensure the convergence in this case.

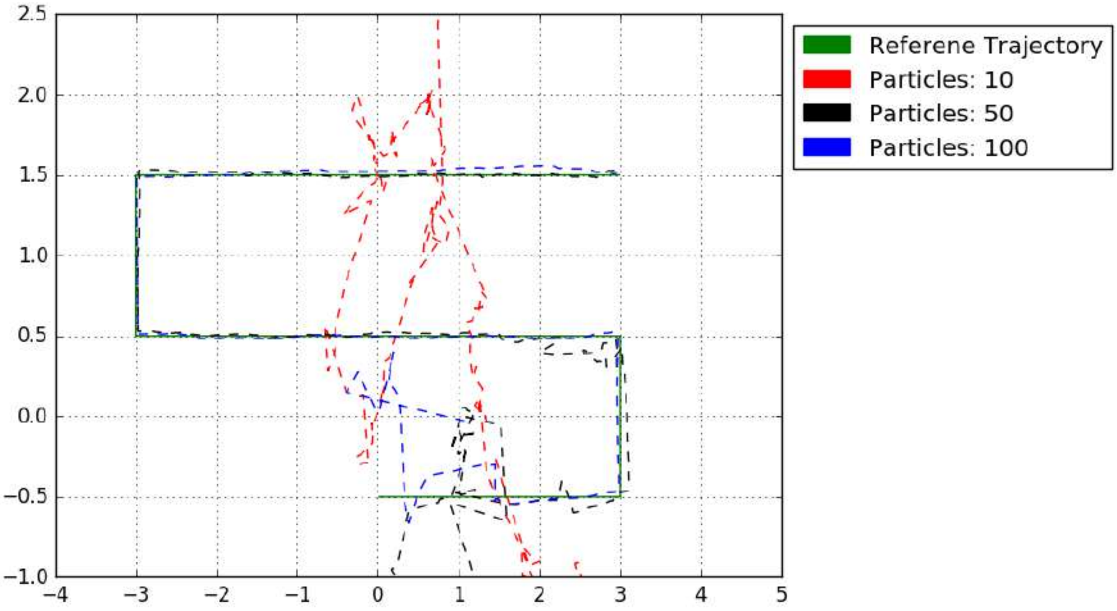
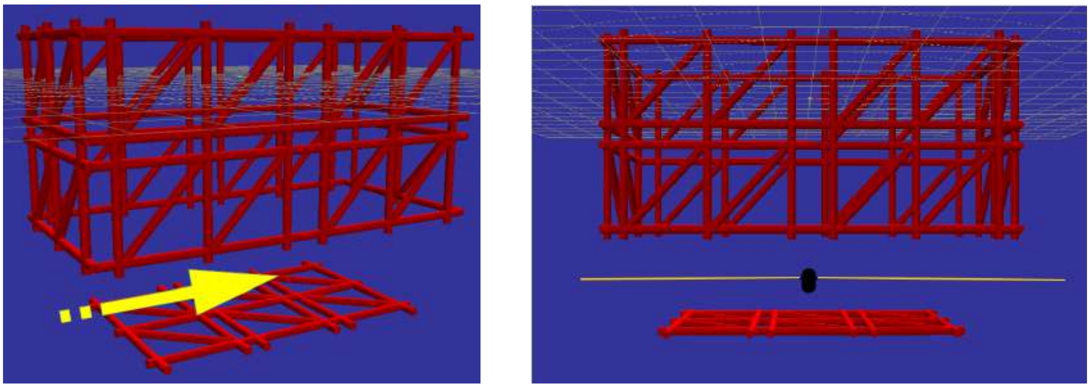


Figure 4.12: Global Localization results for 10, 50 and 100 particles (v=0.1m/s).

4.2.3 Contribution of the method

There are two situations, in the opinion of the author, in which the proposed method surpasses the current localization methods using imaging sonar openly available in the literature. The first scenario is when the vehicle must pass by a horizontal opening, as a gate entrance or a very wide cave, the idea is illustrated in Figure 4.13a. The current range-based methods would have no avaiable information about the environment as the horizontal ray-casting would hit no target at any bearing (Fig. 4.13b). Geometric approaches are also limited in this situation as the line feature based map cannot represent this environment due to the before mentioned 2D limitation.



(a) Trajectory through a horizontal opening and no walls.

(b) Ray-casting methods cannot observe the environment.

Figure 4.13: Horizontal opening scenario where current range methods fail.

The proposed method was capable to cast beams into the map and gather intensity data from both elements above and beneath the vehicle. The wide vertical opening of the sonar allows to sense any target within the $\pm 15^o$ line of sight. The parameters used in the simulation are stated in the Table 4.3. The trajectory and the convergence of the filter can be analysed in Figure 4.14, where the vehicle was hovering at 4m deep.

Table 4.3: Passing through horizontal opening problem

$bel(x_0)$	x_0	$\sigma_{0_{pos}}$	$\sigma_{0_{ori}}$	σ_u	σ_v	σ_r	σ_{int}	σ_{sp_u}	σ_{sp_v}	σ_{sp_r}
$[0,0]$	$[-4,0.5]$	$[3,1]$m	0.3rad	10%	10%	10%	0.1	0.2	0.2	0.02

The second contribution of this work is to refuse the hipothesis that the vertical ambiguity introduced by the wide vertical opening reduces this type of sonar to a 2D sensor. The proposed work does not eliminate the vertical ambiguity within the bin arc, however the method can extract some information about the 3D environment. As the depth sensor provides only a measurement from the water column above the vehicle and not an absolute position along the z-axis, it was introduced another

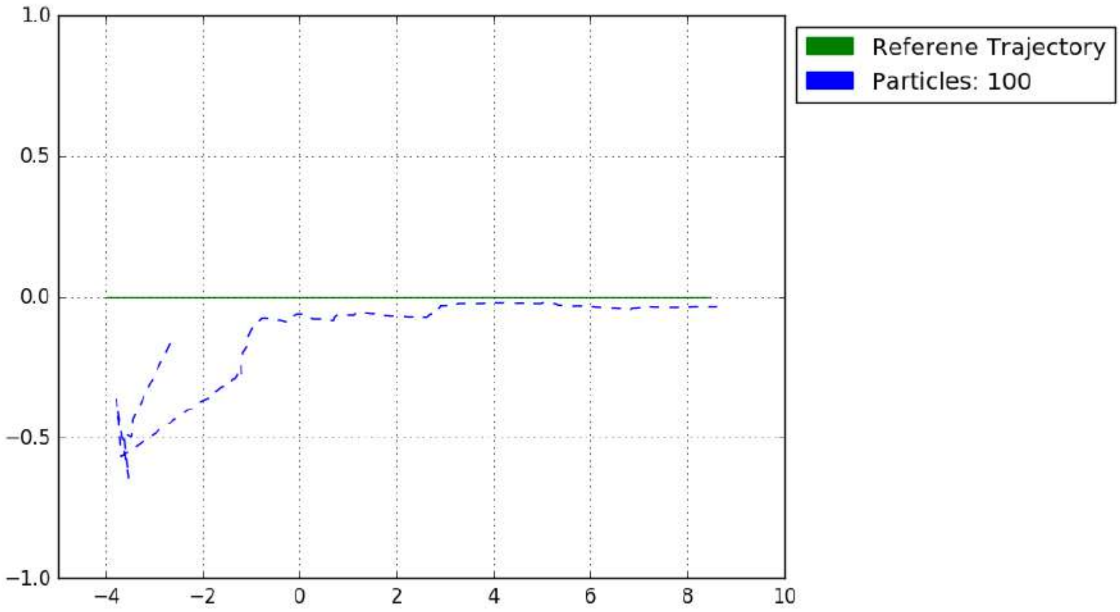


Figure 4.14: Horizontal opening problem results for 100 particles(v=0.1m/s).

dimension to be sampled by the filter, a water column offset w_c. This parameter is important in systems that cannot measure the altitude to the ground, as in the case of the ROV LUMA which does not have a DVL or similar system. The case of the robot ROSA, where tide of the river changes dayly is also well represented under this constraints. The environment where the added dimension w_c was tested is in an open and 3D map, with the presence of a oil and gas manifold and a sunked ship. This map presents a challenge both to the estimation of the x, y, ψ, as well provides a bottom reference. This characteristic is fundamental to the convergence of the water column offset parameter, as it allows the extraction of vertical information based on the distance to the first ground reading. In a situation where there is no horizontal reference (ground or celling) as in the red metal trellis map, the water column initial distribution would remain constant until a suitable reference appears. If the sonar can see inferior or superior limit of the structure for example, the particle outside this range would be eliminated. Figure 4.15 illustrates the initial spread of w_c and Figure 4.16 the final convergence of the filter.

4.2.4 Final Considerations

As in any particle filter implementation, the convergence of the filter depends on the correct tuning of the parameters and it is important to understand the effect of each one to aid in this process. The initilization parameters $bel(x_0), \sigma_{0_{pos}}, \sigma_{0_{ori}}$ are of immediate understanding, as this parameters control the format of the initial belief distribution. Other initialization methods can be applied, as an uniform distribution for example.

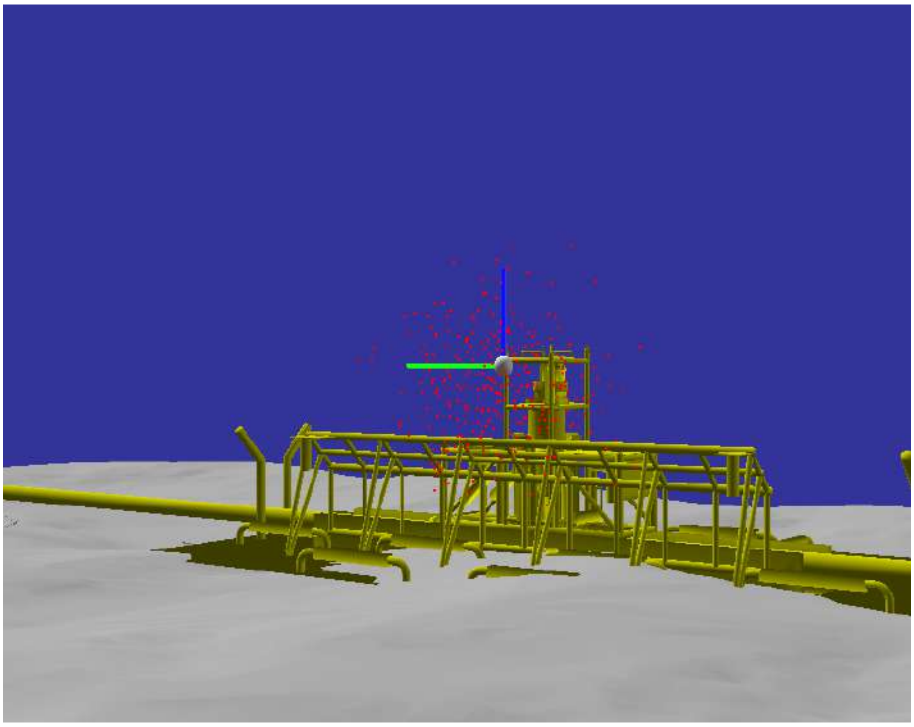


Figure 4.15: Initial spread of the filter in a complex environment.

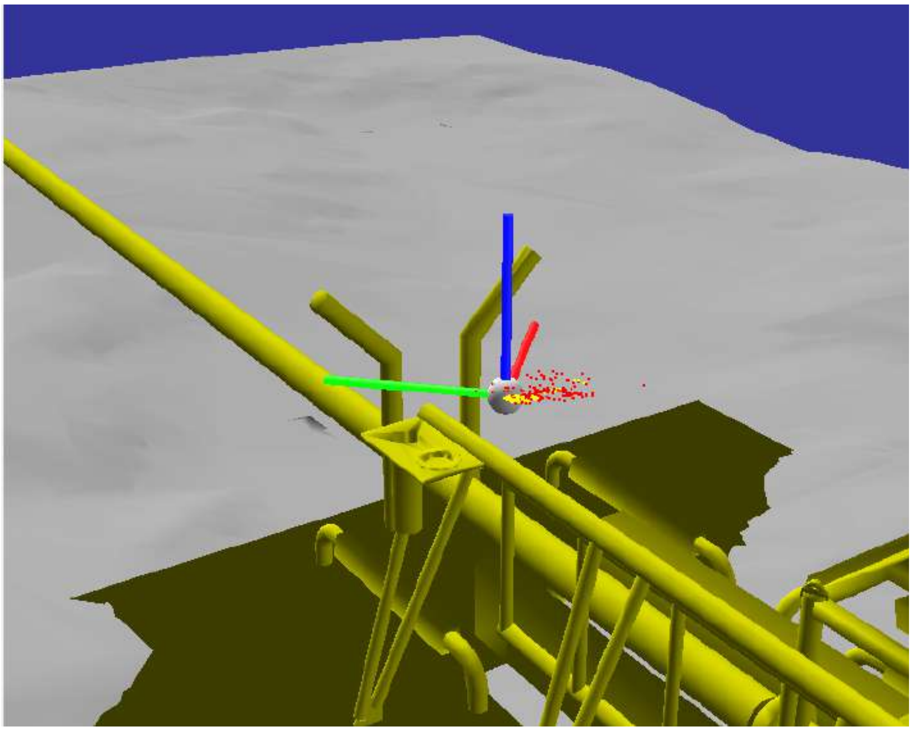


Figure 4.16: Convergence of the filter, inclusive in z in a complex 3D environment.

The motion model parameters can be understood as how the motion prediction can be trusted. If there is access to a good model, and the velocities and inclination input noises are slow these parameters can be decreased. If the incoming noise is relatively big, there is the need to raise this parameters value. A bigger variance in the velocities input is translated as a more normal distributed spread during the update step. This characteristic degrades the estimation, but at the same time allows the filter to realocate particles in regions closer to the true state and avoid a possible divergence.

The degration of the estimation is also introduced by the spread function parameters $\sigma_{sp_v}, \sigma_{sp_r}$, however this effect is only prominent when the maximum weight is low. If one increases the value of this parameters, the filter would be able to recover easily from a kidnapped robot problem, but if the values are set too high, the spread could prohibit the convergence itself.

Finally, the measurement model variance σ_{int}^2 dictates the width of the error function on Eq. 3.54. As its value is decreased, the filter becomes more selective and the convergence speed increases. However if the sensor is considered too accurate this can have serious impact on the filter. To understand this the reader can imagine a deterministic sensor, the measurement distribution function is going to be zero for all most the entire space state. In conclusion, the probability of the samples of the proposal distribution to be exactly on this submanifold of the state space is going to be practically zero and only the particle with a perfect match would have non zero weights assigned and the resampling step becomes ill conditioned.

Chapter 5

Conclusion

5.1 Contributions

This work proposed a localization algorithm using a mechanically scanned imaging sonar based on a new perspective on how to model this sensor. The literature review pointed that the current methods adapt the characteristics of this type of sonar to match the well mature range finder sensor localization theory. However, this introduces simplifications that do not fully explore the potencial of the system and discard useful data.

A mesurement model based on the intensities values of each sonar beam is proposed and validated by simulations using a Particle Filter. This type of filter is well suited for this application as it can handle the global localization problems, where the inital state of the system is unknown, as well handles a wide range of measurement functions.

In order to evaluate the incoming sonar data, sonar simulation was embedded in the algorithm and the wide vertical beam opening was represented to cast simulated beams. This allows the filter to converge in scenarios where the current methods would fail and stands as the major contribution of this thesis. The first scenario is in a situation where there is no reference at the same horizontal plane of the sensor's transducer axis and the system must explore the full vertical field of view of the sensor to be able to observe the map. In addition, the incorporation of the vertical ambiguity in the sensor model allowed the system to extract 3D dimensional information if a horizontal reference is present. An offset depth parameter was sampled by the filter showed convergence in the orthognal dimension of the sensor plane. Considering that the sonar properties incoporated to the model do not invalidate any of the current methods, the algorithm opens possibilities to be exploited, both in the proposed intensity model as well with the range finder model.

5.2 Future Works

According to the main ideas developed in the current thesis, some research topics for future works are:

- Validate the algorithm and the proposed measurement model in a real system. Propose a calibration method to normalize real intensity readings and the sonar simulation. As this work was already developed with this mindset, using the Robotics Construction Kit framework, what allows the evaluation of different systems.

- Compare the proposed intensity measurement model with the range finder model, including an expansion of the range finder model using the imaging sonar beam shape.

- Expand the measurement model to enconpass a Forward Looking sonar. As this is the multibeam counterpart of the mechanically scanned imaging sonar, there is no reason to believe that a study using this type of sensor could not be performed unsing the methods proposed in this work.

- Evaluate the impact of the simplifications made in the simulation of the reference sonar beams, such as multipath, reverberation and sound speed variation. As well compare the results and computational efficiency with other models.

- Incorporate the beam shape properties of imaging sonars to map the environment and propose a Simultaneous Localization and Mapping algorithm using a mechanically scanned imaging sonar. The map representation used in this work, while well suited for a localization only algorithm, as it provides flexibility in representation of arbitrary shapes and high detailment capability, demands some study on how to adapt it to handle a mapping step and how to represent notions of occupancy and uncertanties.

Bibliography

[1] ALBIEZ, J., JOYEUX, S., GAUDIG, C., et al., 2015, "FlatFish - a compact subsea-resident inspection AUV", *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–8.

[2] ALCOCER, A., OLIVEIRA, P., PASCOAL, A., 2007, "Study and implementation of an EKF GIB-based underwater positioning system", *Control Engineering Practice*, v. 15, n. 6 (jun), pp. 689–701. ISSN: 09670661. doi: 10.1016/j.conengprac.2006.04.001.

[3] BATISTA, P., SILVESTRE, C., OLIVEIRA, P., 2009, "A sensor-based controller for homing of underactuated AUVs", *IEEE Transactions on Robotics*, v. 25, n. 3, pp. 701–716. ISSN: 15523098. doi: 10.1109/TRO.2009.2014496.

[4] BELL, J. M., 1997, "Application of optical ray tracing techniques to the simulation of sonar images", *Optical Engineering*, v. 36, n. 6, pp. 1806–1813. ISSN: 00913286. doi: 10.1117/1.601325.

[5] CAPPE, O., GODSILL, S. J., MOULINES, E., 2007, "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo", *Proceedings of the IEEE*, v. 95, n. 5 (may), pp. 899–924. ISSN: 0018-9219. doi: 10.1109/JPROC.2007.893250.

[6] CARNEIRO, R. F., LEITE, A. C., PEIXOTO, A. J., et al., "Underwater Robot for Tunner Inspection: Design and Control", pp. 120–125.

[7] CATALOGUE, H., 2017. "ADCP/DVL Hardware". Disponível em: <http://www.hydrographic-catalogue.com/california/poway/articles/rowe-technologies-adcp-applications-summary-85>. [Online; accessed 12-February-2017].

[8] CERQUEIRA, R., TROCOLI, T., NEVES, G., et al., "Custom Shader and 3D Rendering for computationally efficient Sonar Simulation", .

[9] CHRIST, R. D., WERNLI, R. L., 2014, *The ROV Manual - A User Guide for Remotely Operated Vehicles The ROV Manual A User Guide for Remotely.* ISBN: 9780080982885.

[10] COIRAS, E., GROEN, J., 2009, "Simulation and 3d reconstruction of sidelooking sonar images", *Advances in sonar technology. IN-TECH*, , n. February, pp. 1–15.

[11] CORKE, P., DETWEILER, C., DUNBABIN, M., et al., 2007, "Experiments with Underwater Robot Localization and Tracking", *Proceedings 2007 IEEE International Conference on Robotics and Automation*, , n. April, pp. 4556–4561. ISSN: 1050-4729. doi: 10.1109/ROBOT.2007.364181.

[12] D'AMICO, A., PITTENGER, R., 2009, "A brief history of active sonar", *Aquatic Mammals*, v. 35, n. 4, pp. 426–434. ISSN: 01675427. doi: 10.1578/AM.35.4.2009.426.

[13] DE AGOSTINO, M., MANZINO, A. M., PIRAS, M., 2010, "Performances comparison of different MEMS-based IMUs", *Record - IEEE PLANS, Position Location and Navigation Symposium*, pp. 187–201. ISSN: 2153-358X. doi: 10.1109/PLANS.2010.5507128.

[14] DE BARROS, E. A., FREIRE, L. O., DANTAS, J. L. D., 2010, *Development of the Pirajuba AUV*, v. 43. IFAC. ISBN: 9783902661883. doi: 10.3182/20100915-3-DE-3008.00063.

[15] DEMARCO, K., WEST, M., HOWARD, A., 2015, "A Computationally-Efficient 2D Imaging Sonar Model for Underwater Robotics Simulations in Gazebo", .

[16] DRUMHELLER, D. S., 1998, *Introduction to wave propagation in nonlinear fluids and solids.* Cambridge University Press.

[17] FAIRFIELD, N., KANTOR, G., JONAK, D., et al., 2010, "Autonomous Exploration and Mapping of Flooded Sinkholes", *The International Journal of Robotics Research*, v. 29, n. 6, pp. 748–774. ISSN: 0278-3649. doi: 10.1177/0278364909344779.

[18] FAIRFIELD, N., WETTERGREEN, D., 2008, "Active localization on the ocean floor with multibeam sonar", *Oceans 2008*, pp. 1–10. doi: 10.1109/OCEANS.2008.5151853.

[19] FAIRFIELD, N., KANTOR, G., WETTERGREEN, D., 2007, "Real-time SLAM with octree evidence grids for exploration in underwater tunnels",

Journal of Field Robotics, v. 24, n. 1-2, pp. 3–21. ISSN: 15564959. doi: 10.1002/rob.20165.

[20] FALLON, M. F., KAESS, M., JOHANNSSON, H., et al., 2011, "Efficient AUV navigation fusing acoustic ranging and side-scan sonar", *Proceedings - IEEE International Conference on Robotics and Automation*, (may), pp. 2398–2405. ISSN: 10504729. doi: 10.1109/ICRA.2011.5980302.

[21] FOSSEN, T. I., 1994, *Guidance and Control of Ocean Vehicles*, v. 32. ISBN: 0471941131. doi: 10.1016/0005-1098(96)82331-4.

[22] GU, J.-H., JOE, H.-G., YU, S. C., 2013, "Development of image sonar simulator for underwater object recognition", *2013 OCEANS - San Diego*, pp. 1–6. ISSN: 0197-7385.

[23] GUÉRIOT, D., SINTES, C., 2010, "Forward looking sonar data simulation through tube tracing", *OCEANS'10 IEEE Sydney, OCEANSSYD 2010.* doi: 10.1109/OCEANSSYD.2010.5603867.

[24] HEIDARSSON, H. K., SUKHATME, G. S., 2011, "Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar", *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 731–736. ISSN: 10504729. doi: 10.1109/ICRA.2011.5980509.

[25] HERNANDEZ, E., RIDAO, P., RIBAS, D., et al., 2009, "Probabilistic sonar scan matching for an AUV". In: *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 255–260. ISBN: 9781424438044. doi: 10.1109/IROS.2009.5354656.

[26] HOL, J. D., SCHÖN, T. B., GUSTAFSSON, F., "on Resampling Algorithms for Particle Filters", .

[27] HORNUNG, A., WURM, K. M., BENNEWITZ, M., et al., 2013, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees", *Autonomous Robots*, v. 34, n. 3 (feb), pp. 189–206. ISSN: 0929-5593. doi: 10.1007/s10514-012-9321-0.

[28] HURTOS, N., CUF', X., PETILLOT, Y., et al., 2012, "Fourier-based registrations for two-dimensional forward-looking sonar image mosaicing", *IEEE International Conference on Intelligent Robots and Systems*, pp. 5298–5305. ISSN: 21530858. doi: 10.1109/IROS.2012.6385813.

[29] HURTÓS, N., NAGAPPA, S., PALOMERAS, N., et al., 2014, "Real-time mosaicing with two-dimensional forward-looking sonar", *Proceedings - IEEE*

International Conference on Robotics and Automation, pp. 601–606. ISSN: 10504729. doi: 10.1109/ICRA.2014.6906916.

[30] JAKUBA, M. V., ROMAN, C. N., SINGH, H., et al., 2008, "Long-baseline acoustic navigation for under-ice autonomous underwater vehicle operations", *Journal of Field Robotics*, v. 25, n. 11-12, pp. 861–879. ISSN: 15564959. doi: 10.1002/rob.20250.

[31] JARVIS, R. A., 1983, "A Perspective on Range Finding Techniques for Computer Vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. PAMI-5, n. 2, pp. 122–139. ISSN: 01628828. doi: 10.1109/TPAMI.1983.4767365.

[32] JOHANNSSON, H., KAESS, M., ENGLOT, B., et al., 2010, "Imaging sonar-aided navigation for autonomous underwater harbor surveillance", *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (oct), pp. 4396–4403. doi: 10.1109/IROS.2010.5650831.

[33] JOYEUX, S., ALBIEZ, J., 2011, "Robot development: from components to systems", *6th National Conference on Control Architectures of Robots*, , n. May, pp. 15.

[34] JULIER, S. J., UHLMANN, J. K., 1997, "New extension of the Kalman filter to nonlinear systems". In: *Int Symp AerospaceDefense Sensing Simul and Controls*, v. 3, p. 182. ISBN: 0-7803-6293-4. doi: 10.1117/12.280797.

[35] KALMAN, R. E., 1960, "A New Approach to Linear Filtering and Prediction Problems", *Journal of Basic Engineering*, v. 82, n. 1, pp. 35. ISSN: 00219223. doi: 10.1115/1.3662552.

[36] KAROUI, I., QUIDU, I., LEGRIS, M., 2015, "Automatic Sea-Surface Obstacle Detection and Tracking in Forward-Looking Sonar Image Sequences", *IEEE Transactions on Geoscience and Remote Sensing*, v. 53, n. 8, pp. 4661–4669. ISSN: 01962892. doi: 10.1109/TGRS.2015.2405672.

[37] KHAN, R. R., TAHER, T., HOVER, F. S., 2010, "Accurate geo-referencing method for AUVs for oceanographic sampling". In: *OCEANS 2010*, pp. 1–5. ISBN: 978-1-4244-4332-1. doi: 10.1109/OCEANS.2010.5664570.

[38] KINSEY, J. C., EUSTICE, R., WHITCOMB, L. L., 2006, "A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges", *7th Conference on Manoeuvring and Control of Marine Craft (MCMC'2006)*, pp. 1–12.

[39] KUSSAT, N. H., CHADWELL, C. D., ZIMMERMAN, R., 2005, "Absolute positioning of an autonomous underwater vehicle using GPS and acoustic measurements", *IEEE Journal of Oceanic Engineering*, v. 30, n. 1, pp. 153–164. ISSN: 0364-9059. doi: 10.1109/JOE.2004.835249.

[40] MAKI, T., KONDO, H., URA, T., et al., 2008, "Imaging vent fields: SLAM based navigation scheme for an AUV toward large-area seafloor imaging", *2008 IEEE/OES Autonomous Underwater Vehicles, AUV 2008*. ISSN: 1522-3167. doi: 10.1109/AUV.2008.5290530.

[41] MALLIOS, A., RIDAO, P., RIBAS, D., et al., 2010, "Probabilistic sonar scan matching SLAM for underwater environment". In: *OCEANS'10 IEEE SYDNEY*, pp. 1–8, . ISBN: 978-1-4244-5221-7. doi: 10.1109/OCEANSSYD.2010.5603650.

[42] MALLIOS, A., RIDAO, P., RIBAS, D., et al., 2010, "EKF-SLAM for AUV navigation under probabilistic sonar scan-matching". In: *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 4404–4411, . ISBN: 9781424466757. doi: 10.1109/IROS.2010.5649246.

[43] MALLIOS, A., RIDAO, P., RIBAS, D., et al., 2014, "Scan matching SLAM in underwater environments", *Autonomous Robots*, v. 36, n. 3, pp. 181–198. ISSN: 09295593. doi: 10.1007/s10514-013-9345-0.

[44] MAURELLI, F., KRUPINSKI, S., PETILLOT, Y., et al., 2008, "A particle filter approach for AUV localization". In: *OCEANS 2008*, pp. 1–7. ISBN: 0197-7385. doi: 10.1109/OCEANS.2008.5152014.

[45] MAURELLI, F., PETILLOT, Y., MALLIOS, A., et al., 2009, "Sonar-based AUV localization using an improved particle filter approach". In: *OCEANS 2009 - EUROPE*, pp. 1–9. Ieee, may. ISBN: 9781424425228. doi: 10.1109/oceanse.2009.5278196.

[46] NEWMAN, P., LEONARD, J., 2003, "Pure range-only sub-sea SLAM", *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, v. 2, pp. 1921–1926. ISSN: 1050-4729. doi: 10.1109/ROBOT.2003.1241875.

[47] OCEANOGRAPHIC, A., 2017. "SBL/LBL systems". Disponível em: <http://www.amloceanographic.com/Technical-Demo/USBL-SBL-LBL_2>. [Online; accessed 12-February-2017].

[48] PADIAL, J., DEKTOR, S., ROCK, S. M., 2014, "Correlation of imaging sonar acoustic shadows and bathymetry for ROV terrain-relative localization", *Oceans 2014 - Taipei*, pp. 1–10. doi: 10.1109/OCEANS-TAIPEI.2014.6964351.

[49] PAULL, L., SAEEDI, S., SETO, M., et al., 2014, "AUV navigation and localization: A review", *IEEE Journal of Oceanic Engineering*, v. 39, n. 1 (jan), pp. 131–149. ISSN: 03649059. doi: 10.1109/JOE.2013.2278891.

[50] PETILLOT, Y., MAURELLI, F., VALEYRIE, N., et al., 2010, "Acoustic-based techniques for autonomous underwater vehicle localization", *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, v. 224, n. 4, pp. 293–307. ISSN: 1475-0902. doi: 10.1243/14750902JEME197.

[51] QUIGLEY, M., CONLEY, K., GERKEY, B., et al., 2009, "ROS: an open-source Robot Operating System", *Icra*, v. 3, n. Figure 1, pp. 5. ISSN: 0165-022X. doi: http://www.willowgarage.com/papers/ros-open-source-robot-operating-system.

[52] RIBAS, D., 2013, "Underwater SLAM for Structured Environments Using an Imaging Sonar", v. 1, n. September 2013, pp. 1–36. ISSN: {<}null{>}. doi: 10.1017/CBO9781107415324.004.

[53] RIBAS, D., RIDAO, P., NEIRA, J., et al., 2006, "SLAM using an Imaging Sonar for Partially Structured Underwater Environments", *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (oct), pp. 5040–5045. doi: 10.1109/IROS.2006.282532.

[54] RIBAS, D., RIDAO, P., TARD, J. D., et al., 2007, "Underwater SLAM in a marina environment", *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1455–1460. doi: 10.1109/IROS.2007.4399222.

[55] RIBAS, D., RIDAO, P., TARDOS, J. D., et al., 2008, "Underwater SLAM in man-made structured environments", *Journal of Field Robotics*, v. 25, n. 11-12, pp. 898–921. ISSN: 1556-4959. doi: 10.1002/rob.v25:11/12.

[56] RIDAO, P., CARRERAS, M., RIBAS, D., et al., 2010, "Visual inspection of hydroelectric dams using an autonomous underwater vehicle", *Journal of Field Robotics*, v. 27, n. 6, pp. 759–778. ISSN: 1556-4967. doi: 10.1002/rob.20351.

[57] RUIZ, I. T., DE RAUCOURT, S., PETILLOT, Y. R., et al., 2004, "Concurrent mapping and localization using sidescan sonar", *IEEE Journal of Oceanic Engineering*, v. 29, n. 2 (apr), pp. 442–456. ISSN: 0364-9059. doi: 10.1109/JOE.2004.829790.

[58] SAÇ, H., LEBLEBICIOÊĞLU, K., BOZDAÊĞI AKAR, G., 2015, "2D high-frequency forward-looking sonar simulator based on continuous surfaces approach", *Turkish Journal of Electrical Engineering and Computer Sciences*, v. 23, pp. 2289–2303. ISSN: 13036203. doi: 10.3906/elk-1305-188.

[59] SMITH, S. M., KRONEN, D., 1997, "Experimental results of an inexpensive short baseline acoustic positioning system for AUV navigation", *Oceans '97 Mts/Ieee Conference Proceedings, Vols 1 and 2*, v. 1, pp. 714–720\r1510. ISSN: 01977385. doi: 10.1109/OCEANS.1997.634454.

[60] SOETENS, P., 2006, *A Software Framework for Real-Time and Distributed Robot and Machine Control*. N. May. ISBN: 9056826875.

[61] THORPE, C., DURRANT-WHYTE, H., 2001, *Underwater Robots: Motion and Force Control of Vehicle-Manipulator Systems*, v. 2. ISBN: 3540221085. doi: 10.1007/978-3-540-73958-6_2.

[62] THRUN, S., 2002, "Particle Filters in Robotics", *Proceedings of Uncertainty in AI*. ISSN: 00222275.

[63] THRUN, S., 2002, *Probabilistic robotics*, v. 45. ISBN: 0262201623. doi: 10.1145/504729.504754.

[64] USBL, 2017. "USBL system". Disponível em: <www.thiennampositioning.com/vi/dinh-vi-am-usbl.html>. [Online; accessed 12-February-2017].

[65] VASILESCU, I., DETWEILER, C., DONIEC, M., et al., 2010, "AMOUR V: A Hovering Energy Efficient Underwater Robot Capable of Dynamic Payloads", *The International Journal of Robotics Research*, v. 29, n. 5, pp. 547–570. ISSN: 0278-3649. doi: 10.1177/0278364909358275.

[66] VELKI, 2017. "VKP-011 - Transmissor de Pressão Mini IP65". Disponível em: <www.velki.com.br/produto/pressao/transmissores-de-pressao/vkp-011-transmissor-de-pressao-mini-ip65>. [Online; accessed 12-February-2017].

[67] VICKERY, K., 1998, "Acoustic positioning systems. A practical overview of current\nsystems", *Proceedings of the 1998 Workshop on Autonomous Un-*

derwater Vehicles (Cat. No.98CH36290), pp. 5–17. doi: 10.1109/AUV.1998.744434.

[68] WALTER, M., HOVER, F., LEONARD, J., 2008, "SLAM for ship hull inspection using exactly sparse extended information filters", *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1463–1470. ISSN: 10504729. doi: 10.1109/ROBOT.2008.4543408.

[69] WATANABE, Y., OCHI, H., SHIMURA, T., 2012, "A study of inverse SSBL acoustic positioning with data transmission for multiple AUV navigation". In: *OCEANS, 2012 - Yeosu*, pp. 1–6. doi: 10.1109/OCEANS-Yeosu.2012.6263632.

[70] WELCH, G., BISHOP, G., 2006, "An Introduction to the Kalman Filter", *In Practice*, v. 7, pp. 1–16. ISSN: 10069313. doi: 10.1.1.117.6808.

[71] ZHOU, Z., CUI, J.-H., ZHOU, S., 2007, "Localization for large-scale underwater sensor networks", *Lecture Notes in Computer Science*, v. 4479 NCS, n. 0644190, pp. 108–119. ISSN: 03029743. doi: 10.1007/s11276-009-0210-1.