Eindhoven University of Technology

MASTER

Underwater Acoustic Location Estimation of Flight Recorder Using Onboard Ad Hoc Wireless Sensor Network

Wiepking, Thomas R.B.

Award date:
2022

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Underwater Acoustic Location Estimation of Flight Recorder Using Onboard Ad Hoc Wireless Sensor Network

Master Embedded Systems Thesis Report

Thomas Wiepking student ID 0932578

ALTEN Netherlands Department Technical Software South

Eindhoven University of Technology Department of Mathematics and Computer Science
Interconnected Resource-aware Intelligent Systems

Supervisors:
Tanir Ozcelebi (Eindhoven University of Technology)
Martijn de Greef (ALTEN Netherlands)

Committee members:
Tanir Ozcelebi (Eindhoven University of Technology)
Martijn de Greef (ALTEN Netherlands)
George Exarchakos (Eindhoven University of Technology)

September 28, 2022

Preface

First of all, I would like to thank my company supervisor Martijn de Greef for his excellent supervision throughout this graduation project. His guidance helped me to scope the project and our frequent discussions on complex topics were very insightful.

Subsequently, I would like to thank my supervisor Tanir Ozcelebi for our useful discussions and his continuous interest in the topic.
I also want to thank Tanir Ozcelebi and ALTEN for their flexibility in choice of the project topic. A master graduation project is intense and working on a project that matches your own interests was an important factor for me to stay motivated.

And last but not least, I want to thank Levi Prikken for his personal support, his affection and love.

Abstract

Underwater Wireless Sensor Networks (UWSN) are an upcoming technology for various underwater applications, such as environment monitoring and localization of objects. An important use case for underwater object localization is locating the underwater locator beacon (ULB) of a flight recorder, commonly known as a 'black box', after the unfortunate event of an oceanic flight crash. Current technology for finding this black box uses hydrophone-equipped vessels or autonomous underwater vehicles (AUV). To increase the probability of finding the black box using these technologies, a well-estimated position of the flight crash is required which is not always available after oceanic crashes. Also, these technologies need to travel to the crash location which takes time and complicates the search process. To circumvent these problems, this study presents an alternative approach to locating the sinking black box using an UWSN onboard of the aircraft that starts deploying in the ocean after the detection of an oceanic crash event.

By registering acoustic ping signals of the ULB, the nodes in the network collaboratively estimate the source location using a Time-Difference of Arrival (TDoA) approach. Underwater nodes are used to extend network coverage to deep-ocean levels and communicate their findings by utilizing wireless acoustic modems.

Use of the underwater acoustic medium imposes various challenges such as a slow propagation speed that changes over depth, limited bandwidth and multipath propagation. In this study a set of strategies is introduced to be incorporated in a UWSN that collectively enables localization of the ULB while taking into account the complexities imposed by the underwater acoustic medium. Also a set of communication and scheduling protocols are included that tailor the network hierarchy.

To assess the performance of onboard UWSN-based black box localization, a custom Underwater Acoustic Localization Simulator (UWALSim) is built that utilizes BELLHOP for realistic underwater sound propagation modelling. Furthermore, the simulator can detect signal collisions, simulate noise on the acoustic channel and models energy consumption of nodes in the network.

Several crash scenarios and variations in self-positioning performance have been used to assess the performance of this approach on localizing the black box. These results show that proper self-positioning is crucial to successfully locate the black box. Also partial deployment of the network in the ocean before the wreckage with the black box starts sinking is highly preferable. When satisfying the requirements in self-positioning performance and partial network deployment, the network is able to locate the black box with an error less than 50 meters for at least 75% of the transmitted ULB pings on a 10-second ping interval. Depending on the scenario, experimental results show that this approach provides accurate black box location estimations after 5-25 minutes since the wreckage with the black box started sinking. Therefore, an onboard ad-hoc UWSN helps in increasing the probability of successfully finding the black box by giving accurate location estimations in a short time compared to traditional black box localization techniques.

Contents

Chapter 1

Introduction

1.1 Context and motivation

Flight recorders (commonly referred to as “black boxes”) are an invaluable tool for investigators in identifying the factors behind an accident. Usually these recorders comprise of two individual boxes: the Cockpit Voice Recorder (CVR) and the Flight Data Recorder (FDR). Each flight recorder is equipped with an *Underwater Locator Beacon* (ULB) to aid localization of the recorder. This ULB begins to send acoustic ping signals whenever it is immersed in water. By regulations of the Federal Aviation Administration the ULB must operate for at least 30 days under a maximum water pressure that equivalents up to 6 kilometers depth. The ULB can only be detected under the water surface and the maximum detection range is usually up to 2 or 3 kilometers, depending on several factors including the battery level of the ULB, water temperature and the ambient noise level underwater.[39] Whenever an airplane crashes in the ocean it may become difficult to track back its flight recorders due to the uncertainty of the exact crash site (last known position differs from actual crash position) and the ocean current being inconsistent over time and depth (due to weather or temperature difference). Furthermore it takes time for rescue vessels to reach the presumed crash site. Therefore initial search areas become huge. For the AF-447 flight which crashed in the Atlantic Ocean in 2009, an initial search radius of 40 nautical miles (74.08 km) has been issued centering the last known position, covering a search area of over 17.000 km^2[6].

Current methods in locating a sunken aircraft involve vessels listening to acoustic ULB signals or an autonomous underwater vehicle to enable scanning for signals or obstacles on greater depth. Even though such methods contribute to the localization of flight recorders, it still requires search equipment to travel to the crash site from a remote location. This can take some time, and therefore complicate the search process.

Over the years, *underwater wireless sensor networks* and its applications have become a more trending topic in research. It will not be feasible to station networks of sensors covering all major oceanic flight routes, therefore this work studies an alternative approach where the equipment required to localize a ULB is stationed in the airplane itself. Following an oceanic crash, the equipment is detached from the airplane and starts deploying itself around the crash site to track the sinking trajectory of the ULB of a flight recorder. Since flight recorders are placed in the most reinforced part of the airplane, it has high probability that the flight recorders remain inside the airplane shell and therefore localization of the plane wreckage underwater is more likely to succeed. By having such a solution, the search area for flight recorders can be reduced significantly which increases the probability of successfully obtaining the recordings and remainings of the airplane wreckage.

Many of the state-of-the-art approaches with the goal of locating a ULB work with a two-dimensional localization algorithm assuming that the flight recorder has reached the bottom of the ocean. However, when knowing the sinking trajectory of the ULB one can map the ocean currents on various depth levels at the moment the aircraft was sinking. This information can aid in finding other parts of the aircraft that have been detached from the wreckage that contains the ULB. Also, existing approaches often make simplifications on the oceanic environment that simplifies localization techniques (for instance a constant sound speed). This study aims to find an alternative approach that is able to track the sinking trajectory of the ULB while also providing insight in the implication of certain assumptions on the environment to the localization performance.

1.2 Research questions

To assess whether an onboard wireless sensor network that can operate underwater can solve the problem of successfully localizing the ULB of a flight recorder after an oceanic crash, the following research questions are formulated:

1. When using an underwater wireless sensor network, what is a good strategy for localization of a pulsed underwater source with an error range of at most 50 meters, up to a depth of 6 km?

2. What set of communication protocols must be used in combination with the localization strategy to give a good trade-off between network reliability and lifetime?

3. What battery capacity is required for each node to operate the network for at least 4 hours and what methods can be performed to preserve power usage without breaking localization and network requirements?

1.3 Method and approach

To answer the aforementioned research questions, an extensive background study is done on flight recorders, the underwater medium, source localization by received sounds and underwater networking and communication. Also, the state-of-the-art on localization of pulsed signals is researched. A sensor network is then designed having all features to collaboratively locate the source of a received acoustic ping. This sensor network is then modelled and tested in a custom simulator that takes into account the most important properties and impairments of the underwater acoustic medium. It is demonstrated how certain physical properties of the oceanic environment impact localization and the overall performance of the network.

1.4 Thesis outline

The remaining of this work is structured as follows:

- Chapter 2 consists of an extensive background study on all topics that are relevant for constructing and assessing an underwater network of sensors with the goal of locating an object that emits pulsed signals. ULB properties are discussed, then the properties of the underwater acoustic channel are discussed. After discussing hydrophone technologies, underwater networking and communication concepts are discussed. Finally, localization techniques are discussed, including the pitfalls of each approach.

- Chapter 3 discusses the solution that this study proposes to the problem of locating the ULB in an underwater environment. This includes a detailed specification on the sensor network and how it is deployed, localization techniques and communication techniques between nodes in the network.

- Chapter 4 covers the custom simulator that was built to assess the proposed approach of chapter 3. It discusses all modules involved and provides insight on design decisions and implementation considerations.

- Chapter 5 shows how the proposed approach performs over time by running simulations on various configurations of the network, environment and strategies. It highlights how certain assumptions on the environment and network configurations impact the performance of the network on localization, communication and energy consumption.

- Chapter 6 evaluates the proposed approach with respect to the research questions described in section 1.2. It evaluates this work against the state-of-the-art, it lists the shortcomings and limitations of this approach and includes recommendations on future work.

Chapter 2

Background and related work

2.1 Flight recorder technology

By regulations of the European Organisation for Civil Aviation Equipment, the flight recorders must be able to withstand a number of events related to the impact of an aircraft crash. This includes a large penetration force, impact velocity and crush force. Also, the flight recorder must be able to withstand a fire and underwater pressure up to 6 kilometers depth.

The ULB of a flight recorder emits an acoustic ping signal between 36.5 kHz and 38.5 kHz at a transmission power of 160.5 dB re 1 μPa. This is the unit for acoustic transmission power, which means that the ULB sends pings with a power of 160.5 dB relative to one micropascal (reference pressure underwater). The signal's pulse length is 10 ms and the pulse interval is 1 s.[3]

There exist novel technologies that eject a flight recorder before impact.[21] The recorder is designed to float and can communicate with satellites to update on the position of the flight recorder. Optionally these recorders can be equipped with an array of hydrophones to localize an onboard pinger on the aircraft, and thereby estimating the sinking trajectory. These technologies are however limited to the detection range of the ULB and stop working once the wreckage sinks to a certain depth. Also, it is possible that the wreckage and the floating recorder drift apart due to different ocean currents on the surface and deep sea. As of writing, ejectable flight recorders are applied in some military aircrafts, but ejectable flight recorders that are able to track and report sinking aircraft wreckage have not yet been used in practice.

2.2 Underwater signals

Most of the current underwater communication applications rely on using cables, which are costly for network development. Therefore an increasing interest has been developed in alternative wireless technologies.[14] Radio frequency (RF) channels, which are based on electromagnetic fields, do not propagate well underwater due to the water absorbing the electromagnetic radiation. Even though RF allows for a broad bandwidth, communication ranges are limited to ten meters underwater.

Optical transmission is another communication medium that can be used underwater. However, just like for RF, optical light suffers great from absorption in water. The amount of absorption depends on the wavelength of the optical beam but still the communication range would be limited to at most 100 meters, provided a clear line of sight between transmitter and receiver. Furthermore optical transmission suffers from light scattering due to underwater obstacles (for instance air bubbles) and ambient light noise,

especially in near-surface underwater areas.[14]
Acoustics are an upcoming medium for underwater wireless communication technology due to the great propagation of sound in water. The sound attenuation in water is caused by two components: viscosity (for frequencies above 100kHz) and in seawater the ionic relaxation of boric acid and magnesium sulfate (for frequencies up to 100 kHz). The attenuation of sound underwater is quite low, allowing a communication range of several kilometers. However, since only lower frequencies allow for long-range communication, the bandwidth is limited leading to a data rate of kilobytes per second. Unfortunately, acoustics underwater are subject to a number of environmental impairments making the use of acoustics as communication medium rather complicated.

2.2.1 Underwater acoustic channel

The impairments of the acoustic underwater channel and its properties are discussed.

Sound speed profile

The sound speed underwater depends on several parameters including temperature, salinity and pressure. Since these parameters are changing over depth, the speed of sound also changes over depth. The propagation speed of an acoustic signal is usually averaged on 1500 m/s, but there exist several mathematical models to better estimate the propagation speed given a certain depth. One widely used underwater acoustic propagation speed models is described by Munk [22] and states as follows:

$$\begin{aligned} C &= C_1[1 + \epsilon(\eta + e^{-\eta} - 1)] \\ \eta &= \frac{z - z_1}{0.5 \times B} \end{aligned} \tag{2.1}$$

where C indicates the sound speed at depth z, z_1 indicates the depth with the lowest propagation speed and equals 1300 meters, B indicates the scale depth and equals 1300 meters, C_1 indicates the sound speed at depth z_1 and equals 1492 m/s and ϵ indicates the pertubation coefficient and equals 7.4×10^{-3}. The temperature and salinity of a typical oceanic environment is used in the calculation of ϵ, which makes Munk's propagation speed profile a function of depth z only. Figure 2.1 illustrates the Munk propagation speed profile in the underwater acoustic channel over depth.

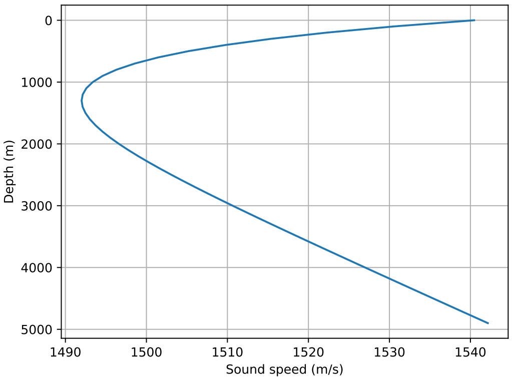


Figure 2.1: Munk underwater sound speed profile.

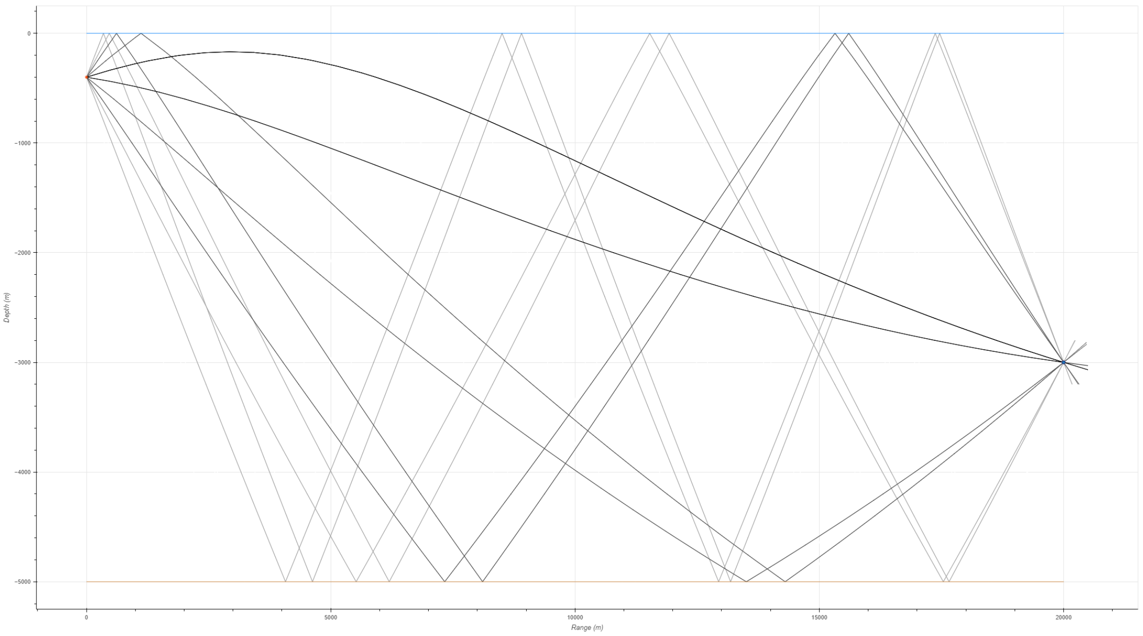


Figure 2.2: An eigenray trace diagram generated by BELLHOP (see section 2.2.3) illustrating the propagation of acoustic waves from transmitter (left) to receiver (right). Notice the reflections on both the surface and bottom and notice the refraction of waves due to the change of speed of sound over depth. Both phenomena are causing multipath interference at the receiver.

Multipath effect

In an underwater acoustic environment, depending on the directivity of the receiver and transmitter and the spreading characteristics of the signal, there may exist multiple paths from transmitter to receiver. Next to the direct path, additional signal paths can be introduced due to surface and bottom reflections. When acoustic waves reflect, the same replica of the transmitted signal may reach the receiver with different attenuations and delays. This may lead to severe inter-symbol interference. Multipath effects can also be introduced due to refractions of acoustic waves, which are introduced due to changes in speed of sound over depth. A ray trace that shows the effects of multipath is illustrated in Figure 2.2.

Doppler effect

If there is a different relative motion between transmitter and receiver, a frequency shift occurs due to the Doppler effect. The change in frequency Δf is given by the following formula:

$$\Delta f = \frac{\Delta v}{c} f_0 \tag{2.2}$$

where Δv is the opposite of the velocity of the receiver relative to the source, c denotes the speed of sound and f_0 denotes the frequency emitted by the source. However, as pointed out by [30] due to the presence of multipath effects and dynamic sound speeds, each individual acoustic wave has a different Doppler shift since each path will have a different angle with respect to the horizon between source and receiver. To minimize the interference that the Doppler effect introduces, several Doppler compensation algorithms tailored to underwater acoustic appliances have been proposed.[32]

Transmission loss

As sound propagates outward from the source in the underwater channel, it loses energy. This loss of energy is important to make estimates on factors like the signal-to-noise ratio (SNR) and in turn for estimations on the probability of a sound being detected by a receiver. The *transmission loss* (TL), sometimes referred to as *path loss* in literature, is measured in decibel and is composed of the following components [19]:

- **Spreading Loss**
  Spreading loss (SPL) is measured in decibells and is calculated using the following equation:
  $$SPL(d) = k \times 10 \log(d) \tag{2.3}$$
  where d is the distance between transmitter and receiver (m) and k is the spreading factor of the signal. For cylindrical spreading, $k = 1$ and for spherical spreading, $k = 2$. When it is assumed that a transmitter sends a signal isotropically (equally omnidirectional), spherical spreading is considered.

- **Absorption Loss**
  Compared to spreading loss, there is little absorption loss on sound underwater. Combining the viscosity and ionic relaxation attenuation components, one can estimate the absorption loss coefficient α, which tells how much a signal drops power over distance (dB/km), Thorp's empirical formula can be used [37]:
  $$\alpha(f) = \frac{0.11f^2}{1+f^2} + \frac{44f^2}{4100+f^2} + 2.75 \times 10^{-4} f^2 + 0.003 \tag{2.4}$$
  where f is the frequency of the signal (kHz). Thorp's formula is based on a generic oceanic environment with a salinity of 25‰, a pH of 8 and a temperature of 4°C. Note that Thorp's formula is only valid for frequencies between 100Hz and 1MHz. Knowing the absorption constant, one can calculate the absorption loss (AL) in decibell:
  $$AL(d, f) = 10 \log(\alpha(f)) \times d \tag{2.5}$$
  where f is the frequency of the signal (kHz) and d is the distance between transmitter and receiver (km).

Noise

There exist four major noise components in the underwater acoustic environment [19]. The formula of each component gives the power spectral density of this noise component in dB re 1 μPa as a function of frequency f relative to 1 kHz:

- **Turbulence noise**
  Noise originating from the disturbance of water, generally occurring in very low frequencies (0.1Hz-10Hz):
  $$\log N_t(f) = 1.7 - 3\log(f) \tag{2.6}$$

- **Shipping noise**
  Noise originating from ship traffic, usually occurring in lower frequencies (10Hz-100Hz). The shipping activity s ranges from 0 to 1:
  $$\log N_s(f) = 4 + 2(s - 0.5) + 2.6\log(f) - 6\log(f + 0.03) \tag{2.7}$$

- **Wind noise**
  Noise generated by the wind, this noise impacts areas near the surface. The wind speed w is measured in m/s.

$$\log N_w(f) = 5 + 0.75w^{\frac{1}{2}} + 2\log(f) - 4\log(f + 0.4) \tag{2.8}$$

- **Thermal noise**
  Noise originating from water molecules, generally occurring in very high frequencies ($>$ 100kHz):

$$\log N_{th}(f) = -1.5 + 2\log(f) \tag{2.9}$$

2.2.2 SONAR equation

Localizing a flight recorder's ULB is an application of passive SONAR, where a system uses the radiated sound from a target to detect and locate it. To make an estimation on whether a receiver is able to capture the ULB signal, the passive SONAR equation can be of use to calculate the Signal-to-Noise ratio (SNR)[27]:

$$SNR = SL - TL - (NL - AG) \tag{2.10}$$

where SL is the source level (dB re 1 μPa at 1m), TL is transmission loss (dB), NL is total noise level (dB) and AG is the array gain (usually 0 dB for small hydrophones).
If the SNR exceeds some detection threshold (DT), it is decided that the signal can be detected. The sound level that exceeds this threshold (i.e. $SNR - DT$) is called the Signal Excess (SE). DT is chosen to such that false positives and false negatives of a signal detection are minimized. DT can be calculated as follows [40]:

$$DT = 10\log\frac{d}{2t} \tag{2.11}$$

where t is the signal transmission time and d is the detection index. The detection index can be calculated based on the desired relation between the probability of detection (PD) and the probability of false alarms (PFA). These relationships are parameterized as d and are outlined in a *ROC curve.* For ULB detection, [2] uses $PA = 0.5$ and $PFA = 1 \times 10^{-4}$, giving a detection index $d = 16$.
Using the detection threshold, a *Figure of Merit* (FOM) is calculated. The FOM indicates the amount of transmission loss where a signal could no longer be detected (i.e. where $SE = 0$). By knowing the FOM, one can plot the TL as a function of range and determine the *detection range* of a signal by intersecting the TL and FOM.

2.2.3 BELLHOP propagation model

BELLHOP is a beam tracing model for predicting acoustic pressure fields in ocean environments. BELLHOP can produce a variety of useful outputs including transmission loss, eigenrays, arrivals and received time-series. It allows for range-dependence in the top and bottom boundaries (*altimetry* and *bathymetry* resp.) as well as a custom sound speed profile.[24]
BELLHOP is used in many literature studies for acoustic underwater propagation modelling and is used in this study to model the blackbox ULB signal. Even though the

problem statement (see section 1.2) involves a 3D Euclidean space, BELLHOP is limited to a 2D space. There exists a 3D extension to BELLHOP called BELLHOP3D and includes (optionally) horizontal refraction in the lat-long plane of acoustic pressure fields.[25] These 3D pressure fields can be calculated by a 2D model simply running it on a series of radials (bearing lines) from the source. This is the so-called *Nx2D* approach. However, this approach neglects the refraction of sound energy out of the vertical plane associated with each bearing line. Such out-of-plane effects can be important when there are significant horizontal gradients in the environment, mostly occurring in strong oceanographic features such as nonlinear internal waves or in areas with strong bathymetric features. This is currently an active area of research. Alternatively, BELLHOP3D also supports *full-3D* mode that does incorporate these features.
Due to the increased computational complexity of a BELLHOP3D full-3D query compared to its 2D predecessor, this study does not incorporate the 3D extension since simulating an event of a sinking ULB with multiple acoustic receivers in the area would lead to undesirably large simulation times. To still incorporate signal propagation in a 3D environment, this study adapts an approach similar to BELLHOP3D's Nx2D approach which is discussed in section 4.4.1.

2.3 Hydrophones

Hydrophones are devices that are able to capture sound in an underwater environment. There exist several types of hydrophones, each having their own sensing mechanisms. The following types of hydrophones are considered: piezoelectric and fiber-optic hydrophones (FOHs). Each type is explained in-depth by [13] and is summarized in this section.

2.3.1 Piezoelectric hydrophones

Piezoelectric hydrophones rely on a piezoelectric sensor that measures pressure changes in the environment. These pressure changes are measured by means of an electrical potential as response to an applied mechanical stress on the ceramic material of the sensor. As an incident acoustic wave impinges on the hydrophone it applies a stress to the material and a voltage is generated. These sensors are however poor at generating current and their produced energy is not enough to transmit the signal over a cable. Therefore, piezoelectric hydrophones are usually battery-equipped. Their sensitivity depends on the frequency that needs to be recorded. The directivity of a hydrophone depends on the shape of the sensor. Cylindrical hydrophones are fairly directional, while spherical hydrophones have a radiation pattern that is close to omnidirectional. Since many hydrophones are often used in deep-water applications, the piezoceramic shells are filled with material that is rigid enough to prevent collapse under high pressure, whilst also providing some damping to broaden the resonance of the piezoceramic element. This makes piezoceramic hydrophones suitable to use in deep ocean water.

2.3.2 Fiber-optic hydrophones

Fiber-optic hydrophones rely heavily on the transduction mechanism. The sensing mechanism is based on the detection of acoustically induced changes in the intensity, phase, refractive index or polarisation of light either guided within an optical fiber (intrinsic FOH) or interacting with an optical sensor located at the distal end of the fiber (extrinsic

FOH).
Generally speaking fiber-optic hydrophones are very sensitive and energy-efficient. They do not require a source of power to transduce sensor readings. Practical applications of fiber-optic hydrophones on deep-sea level is limited. Plotnikov et al. [23] present a method for detection and localization of sound sources underwater using a hydrophone array containing 6 fiber-optic hydrophones and use a fiber-optic cable of 100 meters. By making use of a multiplexing technique on the fiber by modulating laser pulses, they are able to read multiple hydrophone values via a single transducer. Due to the fragility of fiber-optic hydrophones, deploying must be done with great care and reaching depths over 100 meters may damage the sensor due to high pressure working on the hydrophone. Meng et al. [20] published a survey on the recent progress in fiber-optic hydrophones. They mention that the self-noise on FOHs consists mostly on noise from very low frequencies (< 100 Hz). It is also mentioned that multiplexing techniques exist to enable the use of multiple hydrophones on a single fiber pair. Fiber-optic vector hydrophones are a special type of FOH where several orthogonally mounted fiber accelerometers are mounted to suppress isotropic noise.[20] Their detection range is increased and it is reported that these type of hydrophones can resist up to 66 MPa, enabling them to be used up to 6 km depth under water. However, as of writing practical studies that support this claim have not been found. Fiber-optic systems can be either be repeater-based or repeaterless. Repeater-based systems can reach distances up to almost 1000 km, whereas repeaterless systems can reach a distance of up to 120 km.

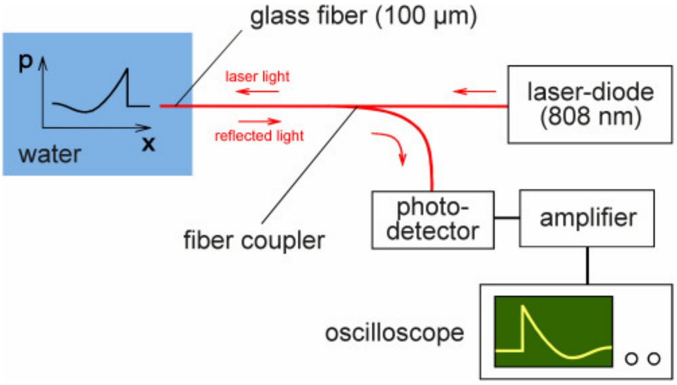


Figure 2.3: Operation principle of the FOPH-2000 by RP Acoustics, a commercially available fiber-optic hydrophone.[1] This hydrophone is based on the Eisenmenger FOH where the mismatch in refractive index between two lights due to acoustically induced changes is used to measure underwater sound.

2.3.3 Combination of piezoelectric hydrophones and fiber-optics

Heerfordt et al.[12] have combined piezoelectric hydrophones and a fiber-optic channel to listen to sperm whale sounds and localize the origin. They have developed a system that uses the fiber-optic connection to transfer digital readings from an analog-to-digital converter attached to an array of piezoelectric hydrophones. Piezoelectric hydrophones are more prone to high pressure, which allows for recording sounds on larger depths, in this study up to at most 950 meters. Since the signal conversion and hydrophones require power to operate, batteries are included in the underwater module. Due to the low temperatures under water, the batteries would only operate up to 6 hours (assuming a temperature of $0°C$ at maximum depth). The data was processed on the boat floating on the surface, to save computation energy in the underwater module. Since the system was quite fragile, it required a special deployment system on the boat whereas deploying would take 1 to 1.5 hours to reach maximum depth.

2.4 Underwater Networking and Communication

In this section the main concepts of underwater networking are discussed, providing that the network operates on acoustic communication between underwater nodes. Depending on the acoustic channel, data loss can occur during transmissions. To limit these losses, acoustic modems can incorporate a number of countermeasures. Relevant countermeasures and involved communication concepts are also discussed in this section.

2.4.1 Underwater Wireless Sensor Network concepts

Han et al. [11] survey numerous techniques to accomplish a reliable underwater wireless sensor network by means of acoustic communication. Most of the existing underwater wireless sensor network layouts, including the network in this study, consist of the following node types:

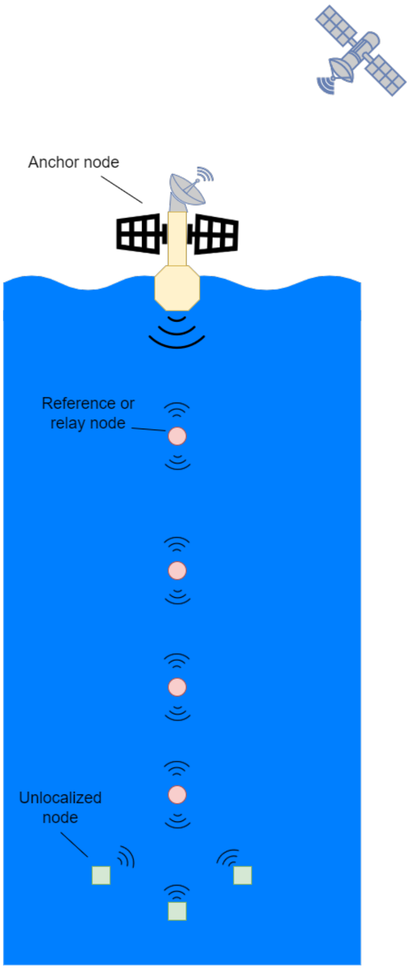


Figure 2.4: Basic architecture of most novel UWSN, having an anchor node, reference nodes and unlocalized nodes.

- Anchor nodes: nodes that can obtain their exact location. Usually these nodes are floating on the water surface and can therefore make use of existing terrestrial localization techniques such as GPS. Their enclosure is similar to that of a buoy, which due to its buoyancy properties will remain on the water surface. These nodes are equipped with acoustic transceivers to establish contact with nodes that are underwater. Several anchor nodes can exist in a network, and their internal clocks can be assumed to be synchronized. It can also be assumed that anchor nodes are able to establish an internet connection by means of for instance satellite communication. Anchor nodes are battery-powered but can make use of other power sources (e.g. solar panels or wind turbines) to extend its operation life.

- Reference (or relay) nodes: nodes consisting of acoustic transceivers and can be equipped with a pressure sensor to measure depth. Their purpose is to relay data packets from and to underwater areas that the anchor nodes cannot reach directly. Optionally, their location is estimated based on information obtained from the anchor node such that reference nodes can aid localization of nodes that yet need to be localized.

- Unlocalized nodes: nodes that perform sensing tasks underwater. They are equipped with acoustic transceivers and an optional pressure sensor to measure depth. The aim of the network is to relay sensor data from unlocalized nodes to the anchor node and be able to localize these nodes such that their measurements can be mapped to a location.

2.4.2 Signal modulation

Modulation is the process of converting data into signal waves by varying one or more properties of a periodic waveform (carrier wave) with a separate wave (modulation signal) that contains the information to be transmitted. Over time, various modulation techniques have been proposed for an acoustic modem. Modulation schemes can be classified as coherent or incoherent modulation schemes. Most early underwater acoustic communication systems use incoherent modulation schemes, where no carrier phase information is required for receiving signals. Receivers are able to extract data packets from signals without pre-knowledge from the transmitter. The biggest advantage of incoherent schemes is their algorithmic simplicity. However due to environmental impairments, systems used in channels with little boundary action (e.g. vertical links in deep-water) would not be able to use incoherent modulation methods.[5] An example of a much used incoherent modulation technique is Frequency-Shift Keying (FSK), where two frequencies are used to modulate data (e.g. frequency $F1$ represents bit 0 and frequency $F2$ represents bit 1). As opposed to incoherent schemes, coherent schemes can encode data using different phases of a given sinusoid allowing for a larger data throughput on the same channel bandwidth (so-called *higher-order modulation*). In order to use coherent schemes, knowledge on the received signal's frequency and phase on the receiver's side is required.

One widely used coherent modulation scheme is *phase-shift keying* (PSK) which modulates data by changing the phase of a constant frequency carrier wave. Higher-order modulation can be achieved by increasing the *symbol alphabet size m*. Each symbol corresponds to a phase relative to the carrier wave and the m phases are equally spaced around a circle. Figure 2.5 shows the constellation diagram of *Binary phase shift keying* (BPSK, $m = 2$) and *Quadrature phase shift keying* (QPSK, $m = 4$). Each phase has a decision boundary that is used for demodulating the (noised) signal. As long as the received phase relative to the carrier do not cross the decision boundary, a signal can be correctly translated (demodulated) to the appropriate data value.
Naturally, the more symbols in a constellation (i.e. more phases mapped on the circle), the more data can be transmitted while keeping the same channel bandwidth. Of course, a higher symbol size narrows down the decision field and therefore decreases the error-margin of a signal. In noisy channels this could lead to signals crossing decision boundaries and therefore being mapped to the wrong value. Hence, depending on the SNR and the required data throughput, an appropriate symbol size should be chosen.

2.4.3 Packet acknowledgment

To prevent data loss, each data packet that was transmitted should be acknowledged by the intended receivers. Therefore after receiving a packet, the receiver should transmit an *acknowledgment* (ACK) packet. If the acknowledgment is not received by the transmitter, the data packet is retransmitted. Even though this approach aids in correct data transmission, due to the long propagation delay in the underwater acoustic channel, frequent retransmissions and ACK returns give much overhead.

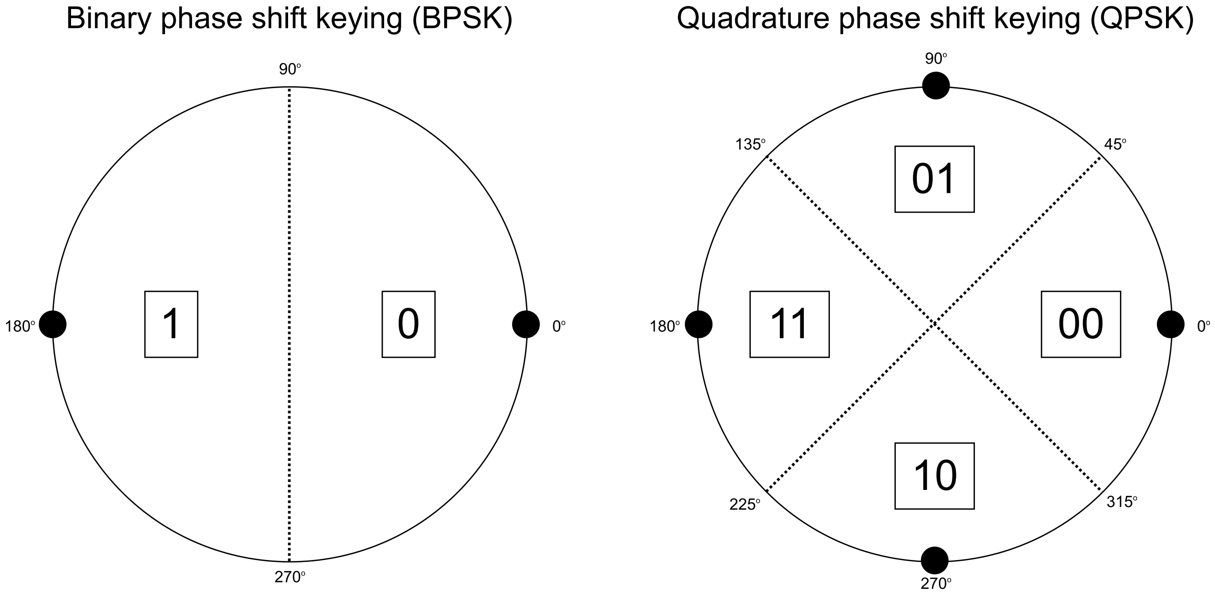

Binary phase shift keying (BPSK)
90°
180°
1
0
0°
270°
Quadrature phase shift keying (QPSK)
90°
135°
45°
01
180°
11
00
0°
10
225°
315°
270°

Figure 2.5: Constellation diagrams of BPSK and QPSK. Each black dot represents a phase relative to the carrier wave. The circle is divided into slices, where each slice represents a decision field. Received signal points will be decoded to the data value corresponding to the decision field it lies in.

2.4.4 Forward Error Correction (FEC)

Due to noise on the channel, some symbols in a transmission may be demodulated incorrectly and cause corruption on (part of) a data packet. To prevent a (costly) retransmission, one can choose to add *Error Correction Codes* (ECCs) to the packet in advance. In the case of an incorrect symbol, depending on the number of ECCs and FEC algorithm, the incorrect symbol can be detected and repaired. However, by adding ECCs to a packet the size of the packet increases and therefore naturally increases the probability of having failed symbols in a packet. In a very noisy channel, FEC might deteriorate a transmission's reliability.

One widely used FEC algorithm is *Reed-Solomon error correction* [26]. Applications using this technique include compact discs, QR codes, satellite communication protocols and storage systems such as RAID 6. Reed-Solomon codes have the following error-correction relation:

$$2E + S \leq n - k \tag{2.12}$$

with n being the block length, k being the number of data symbols in a block, E the number of errors in a block and S the number of erasures (known error locations) in a block. A commonly used Reed-Solomon code is $(n, k) = (255, 223)$, meaning there are $255 - 223 = 32$ error correction symbols in a block. This code is therefore capable of correcting up to 16 symbol errors or 32 erasures in a block, or a combination of the two while still satisfying equation 2.12.

2.4.5 Channel capacity

The *Channel Capacity* C (bits/s) is the tightest (theoretical) upper bound of the information rate of data that can be communicated over a channel. The Nyquist rate states that the number of independent pulses that can be put on a channel per time unit is twice the bandwidth of the channel. Hartley used this analogy to state a relationship on the channel capacity, channel bandwidth and m different pulse levels (symbol alphabet size) in his theorem, assuming a noise-free channel [9]:

$$C = 2B \log_2(m) \tag{2.13}$$

where B is the channel passband bandwidth (Hz). Shannon took this a step further to assume a channel that is subject to additive white Gaussian noise (AWGN).[31] The Shannon-Hartley theorem states the theoretical upper bound of the channel capacity given a channel bandwidth and a SNR. The theorem is described in the following equation:

$$C = B \log_2(1 + \frac{S}{N}) \tag{2.14}$$

where B is the channel passband bandwidth (Hz), S is the received signal power (W) and N is the noise power (W). The number of pulse levels m implicitly assumed here is a theoretical upper bound that in practice is not achieved by current modulation schemes, but can be used as a metric in developing a modulation scheme. This theoretical upper bound can be determined in the following equation:

$$m = \sqrt{1 + \frac{S}{N}} \tag{2.15}$$

2.4.6 Signal collision avoidance

Due to the low propagation speed of sound underwater, there exists an uncertainty of the global state of the channel. This is called the *space-time uncertainty* [36]. This uncertainty occurs in all communication, but gets significant in networks with a large propagation delay. This means that not only a transmitter should have exclusive access to the medium, but also propagating sound waves from previous transmissions should be considered to prevent collisions at the receiver.

Van Kleunen describes a scheduling approach designed for underwater acoustic sensor networks that rule out the following types of conflicts [15]:

- TX-TX conflict: two transmissions are scheduled at the same time from the same source.
- TX-RX conflict: a node cannot receive a packet while transmitting a packet.
- RX-RX conflict: two transmissions with the same receiver arrive within the same time frame.
- RX-interference: while receiving a transmission, another transmission with a different destination interferes with the current transmission.

Assume transmission tasks δ_i and δ_j where $i < j$. Each transmission task has a *start* time, a *duration*, a *source* (src) and a *destination* (dst). Function $T(a, b)$ denotes the propagation time from a to b. This gives the following constraints on transmission start times [15]:

$$\text{Given } j \text{ for all } i < j : \begin{cases} \delta_j.start \geq \delta_i.start + \delta_i.duration & \text{if } \delta_i.src = \delta_j.src \\ \delta_j.start \geq \delta_i.start + \delta_i.duration + max(& \text{if } \delta_i.src \neq \delta_j.src \\ \quad T(\delta_i.src, \delta_i.dst) - T(\delta_j.src, \delta_i.dst), & \\ \quad T(\delta_i.src, \delta_j.dst) - T(\delta_j.src, \delta_j.dst)) & \end{cases} \tag{2.16}$$

2.5 Underwater localization techniques

Since conventional GPS receivers do not work underwater, alternatives based on acoustic communication must be used to localize underwater nodes. Two categories of localization are distinguished: pulsed source localization and bidirectional communicative node localization. While most localization schemes can be theoretically applied to both categories, their practical application for pulsed source localization often is constrained and require certain assumptions in order to succeed. Underwater localization schemes from literature that are discussed in this section assume that the depth of a source is known (either by geographic maps or onboard depth sensors), and therefore their schemes are focused on 2D localization unless mentioned otherwise.

2.5.1 Pulsed source localization

Pulsed source localization refers to the localization of a source node that is only capable of transmitting fixed-length signals at a fixed frequency on a fixed interval. The ULB is categorized as a pulsed source. For convenience it is assumed that the transmitted signals spread isotropically, meaning that sound rays propagate evenly over all directions. Since the transmitted signal does not contain any information about the node, localization must be performed only on information that can be derived by the receiver. Various works exist on this topic, where each builds upon one of the following localization methods: Angle of Arrival, Time of Arrival and Time Difference of Arrival. Their basic approach and tradeoffs are discussed. Afterwards, some proposed improvements from literature are mentioned.

Angle of Arrival

Angle of Arrival (AoA), sometimes referred to as Direction of Arrival (DoA), estimates the angle between a receiver and a transmitter by exploiting the time differences of signal arrival at several hydrophones on a single sensor arm. It assumes that the receivers are far enough away from the source to allow the spherical wave propagation being approximated by planes.[16]. By estimating the speed of sound underwater, fixing the inter-receiver distance and by assuming clock-synchronization of the receivers, the angle of the direction of the source to each receiver can be estimated. By having three or more receivers on a sensor arm, localization can be performed by calculating the distance and angle towards the transmitter from each receiver. However, Sun et al. [35] mention in their AUV-aided flight recorder localization study that the precision of DoA-based methods highly depend on an accurate angular measurement precision, which is difficult to get over larger distance. Since the distance between the flight recorder and hydrophone receivers can be large, they reject the use of DoA for their approach.

Time of Arrival

Time of Arrival (ToA) is a simple technique to implement and assumes that each receiver's exact location is known. Also, it is assumed that the clocks of the transmitter and receivers are synchronized, and the periodicity of the signal being transmitted is known. When the signal arrives at a receiver, the receiver can calculate its distance from the transmitter by using the speed of sound. When performing this technique at two or more locations in the grid, a location can be estimated using circular intersection. Unfortunately, this technique requires that the start time of the signal transmission is known to the receiver and this is not always possible when locating a pulsed source. Usually the receivers arrive later than the first signal has broadcast.

Time Difference of Arrival

Time Difference of Arrival (TDoA) also assumes that the receiver knows its exact location, but it is no longer required to assume clock synchronization between receiver and transmitter and the time that the signal was broadcast. It does however assume that all receivers have synchronized clocks. To geometrically determine the source location, between each pair of receivers R_i and R_j a hyperbola is drawn having as foci the location of R_i and R_j respectively. The length of the transverse axis depends on the TDoA between R_i and R_j, multiplied by the speed of sound underwater. When performing this technique between at least two distinct pairs of receivers (hence having at least three receivers), the hyperbolic intersection represents an estimated location of the source.
Next to the aforementioned two-dimensional approach, Wong et al. [28] propose an alternative geometric approach to pulsed localization for 3D environments. Instead of a 2-dimensional hyperbola, this approach generates a quadric hyper surface hyperboloid by the method of volume of revolution on the x-axis.
Also, the source location can be numerically estimated by calculating a least squares solution to a system of equations as demonstrated by [18]. If TDoA is applied to a periodic signal in a low-propagation medium, it can become unclear to a receiver if it received signal x or if it received signal $x - 1$, $x - 2$ and so on. This problem exists for the flight recorder case and is discussed further in section 3.2.1.

Advanced pulsed source localization techniques

Sun et al. [35] propose a method where an Autonomous Underwater Vehicle (AUV) is used to track back a sunken flight recorder. According to their research, the signal period of the ULB cannot be assumed to be known due to very low temperature underwater and increased pressure. Therefore, they propose a Second-Order Time Difference of Arrival (STDoA) approach. STDoA is defined as the difference of TDoA. If three signal samples are selected uniformly and are strictly consecutive, it allows elimination for the signal period variable in their equation. When using four signal samples, two Second-Order Time Difference of Arrival equations can be formed, each indicating a curve. The intersection of the two curves indicates the estimated position of the flight recorder. According to simulations performed by Sun et al. on their approach, a circular path of the AUV around the actual location of the flight recorder gives the highest localization precision. Within a 2×2 km area it is guaranteed that the localization error is below 20 meters (under good noise circumstances). However, STDoA heavily relies on the signal arrival time and the accuracy of the AUV's own location coordinates. To optimize performance, a tracking algorithm is used based on a particle filter. It uses historic information to increase localization precision and it filters out outliers.

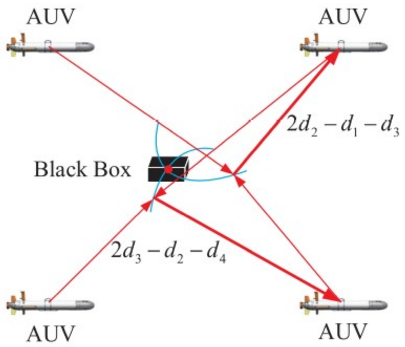


Figure 2.6: Illustration of the STDoA localization approach from [35].

Skarsoulis et al. [33] propose a localization method using a pair of spherical piezoelectric hydrophones towed under a marine vessel. Their approach is tested in the Eastern Mediterranean Sea off the north and south coast of Crete, in shallow and deep water respectively. They used a pinger to transmit signals and later they tested their approach on localizing sperm whales, which emit click-sounds between a frequency of 10kHz and 30kHz. Since sperm whale clicks are not periodic, the study exploits ray theory by measuring the TDoA between direct and surface-reflected rays arriving at the array of hydrophones. By using a Bayesian approach to localization, over multiple iterations the precision on localization estimations must be increased. To accommodate errors in hydrophone depth, sound speed at certain depths and TDoA measurements, they introduced random variables for each of these. A Jacobian matrix is created that expresses the sensitivity of the TDoA to changes in the source ranges and hydrophone depth ranges (since they may move). In an iterative inversion scheme, a first guess is made for the horizontal distances from the source to the hydrophones. This is used as a linearization reference (including reasonable errors based on reference values). The result of the first inversion is then used as new linearization reference. In subsequent iterations, the standard deviations for the distances from the source to the hydrophones are gradually relaxed in order to remove the corresponding constraint. This continues until convergence is established (further iterations do not influence observation errors). If the horizontal location of each hydrophone is known, the source horizontal location can be estimated using triangulation or a DoA estimation approach.

When using a TDoA approach on direct and surface-reflected rays, it is important that the time difference is large enough to distinguish both paths. To properly distinguish both signals, a TDoA of at least 7ms is required which is achieved by placing the hydrophones at a certain minimum depth. The longer the range of their localization system, the deeper the hydrophones must be placed. A number of experiments have been performed on different transmitter and hydrophone depths. From their experiments, it turns out that this approach is very sensitive to the roughness of the sea and hydrophone orientation and generally has a large error in range estimation. In deep water, range errors vary from 150m RMS (d = 1km) up to 1km RMS (d = 3km) where d is the actual distance between the transmitter and the hydrophone array. Even though individual localization results may be of limited use due to the large errors, the situation can be improved by averaging. Also, care must be taken to distinguish direct and surface-reflected arrivals from bottom-reflected arrivals in shallow water. Skarsoulis et al. conclude that self-noise of the hydrophones is the main reason for limitations on range.

2.5.2 Bidirectional communicative nodes

Bidirectional communicative node localization refers to localization of underwater nodes where each node (including the unlocalized nodes) can aid with extra information by means of data exchanges. This could enable clock synchronization and improve localization by utilizing onboard sensors of the unlocalized node.
Diamant et al.[7] describe a sequential algorithm for underwater node localization that assumes no time synchronization and known propagation speeds upon starting the localization procedure. Their algorithm is called Sequential Time-Synchronization Localization (STSL). The intuition behind this approach is the use of relative speed and direction information available at the unlocalized node to compensate for node mobility. In doing so, three or more range measurements obtained at different times and locations can be combined for 2D localization. Depth is assumed to be known by measurements in the sensors. This algorithm relies on periodic packet exchange between the network nodes, to reduce communication overhead. The algorithm consists of two major steps, all performed in a localization window of a fixed amount of time.

1. Time synchronization step: the objective of this step is to provide estimates of the propagation delays for all packets. This is accomplished by a two-way packet exchange, obtaining the local times of receiving and local time of transmitting a packet. Due to permanent motion the propagation delay may not be equal over packets. To tackle this, Diamant et al. apply a quantization mechanism to allow for differences in the propagation delay of separate packets and to enable time-synchronization per anchor node making use of the ongoing network communications. Clock skews and offsets are estimated, after which the propagation delays are estimated.

2. Localization step: The objective of the localization step is to estimate the unlocalized node's coordinates at the end of the localization window. This method can be extended by an iterative refinement, or a self-evaluation of the localization performance. For the latter, Diamant et al. propose a binary test for self-evaluating localization accuracy. This test can be used to finetune parameters, such as the localization window.

To allow multiple nodes utilizing the same communication medium, Time Division Multiple Access (TDMA) is used having time slots of 60 seconds. Sea trial experiments show that the localization estimation error significantly increases with small errors in propagation speed estimation, giving a 50m localization error for a 10m/s propagation speed error. Without propagation speed errors, localization errors are restricted to 15m. The work of Diamant et al. is yet to be extended by a tracking mechanism to continuously localize nodes, where newly localized nodes can also serve as anchors. Their approach, simulation and trials focus on maximizing and scaling on horizontal range. It is not known how well their approach works for networks that mainly consist of vertical links.

Chapter 3

Approach

3.1 Network setup

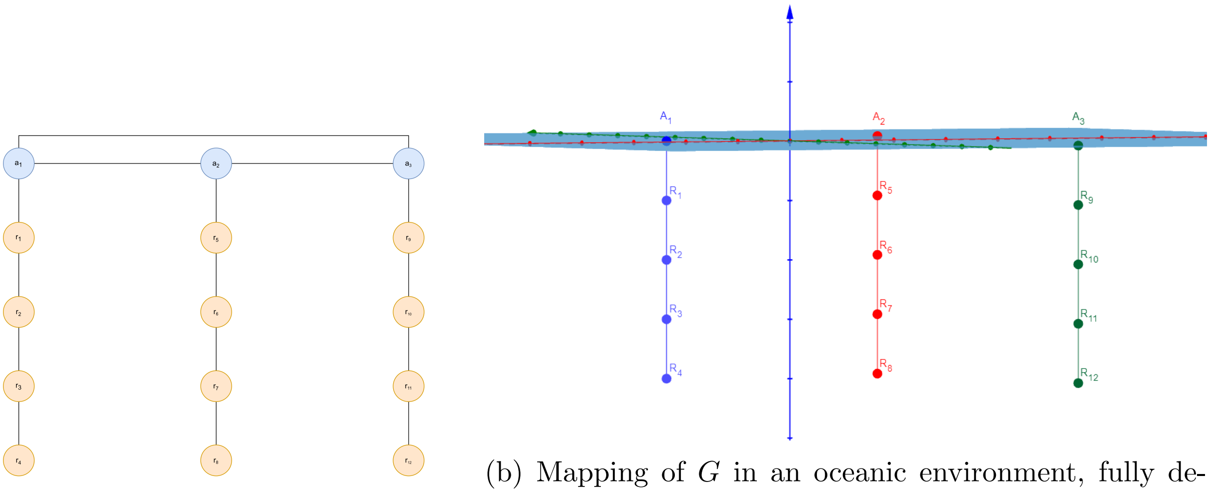
For this study, a network is created that is able to operate ad-hoc in any deep-water environment. It must be able to capture signals up to 6 kilometers depth. The network contains *anchor nodes* and *relay nodes* as introduced in section 2.4.1.

Considerations

Since the acoustic range of the ULB is between 2 and 3 kilometers, placing acoustic receivers just below the water surface will not be sufficient to localize wreckages once they reach deep-sea. Since ocean currents can change in deep-sea areas it cannot be assumed that sinking particles continue on their former trajectory. One possible way to extend a network's depth range is extending the length of the hydrophone cable attached to a surface-node such that acoustic receivers can reach a sufficient depth to observe acoustic signals from the ULB. However, powering a hydrophone via a copper cable of this length would require too much power and brings practical difficulties. Another option would be to use batteries for powering deep-water hydrophones and using a cable connection just for data transfer, which does not require much current over the long cables. There exists technology that can extend wired serial connections up to 1 km. However, the downside of using this solution is that a single defect in the data cable would break the entire network. Also, it complicates the design of the underwater receivers to remain waterproof at large depths. Even though fiber-optic cables or -hydrophones could mitigate most practical problems, these cables and sensors are very fragile. They will take too much damage from the crash and they are unlikely to survive deep-ocean pressure. As of writing, are no practical studies supporting the successful use of fiber-optic hydrophones in deep water. Therefore, to increase detection range to large depths, this network makes use of a chain of relay nodes that communicate over a wireless channel. These relay nodes can listen for ULB signals and communicate their findings to anchor nodes, if necessary through other relay nodes. If a node in the chain breaks, wireless transmissions could still be received by other nodes, making the network more robust.

Deployment

All equipment that is required by this network is onboard of the aircraft. When an immanent oceanic crash is detected, the network equipment is ejected from the aircraft. There exist mechanical applications to detect an oceanic crash event and shoot a flight recorder from the aircraft.[21] A similar approach can be applied to anchor nodes part of this sensor network. It is here assumed that the ejected equipment is reinforced such that it survives the dispatching. Relay nodes reside inside a compartment of an anchor



(a) Graph G

(b) Mapping of G in an oceanic environment, fully deployed

Figure 3.1: Example network G with $o = 3$ and $q = 4$

node. This way, they can be protected from the crash. When an anchor node detects that it has contact with water, it can start releasing the relay nodes in its compartment one by one. Over time, when all relay nodes have been dispatched from all anchor nodes, the network will look like how it is illustrated in Figure 3.1b.

Formal representation of the network

The network can be represented by an undirected graph $G = (V, E)$ with $V = \{v_1, ...v_i, ...v_n\}$ representing the set of nodes with $n = |V|$ and $i \in [1, n]$ and $E = \{e_1, ...e_j, ...e_m\}$ representing the edges with $m = |E|$ and $j \in [1, m]$. An *edge* is a connection between a pair of nodes that represents communication capability between this pair of nodes. Nodes that share an edge are called *neighbors*. The set of nodes V are partitioned into subsets A, R such that $V = \{A \cup R\}$ and $\{A \cap R\} = \emptyset$. Each subset is defined as follows:

- $A = \{a_1, ...a_k, ...a_o\}$ represent anchor nodes with $o = |A|$ and $k \in [1, o]$. Vertices in A are *fully connected*, i.e. $\forall_{a_1,a_2 \in A}$ with $a_1 \neq a_2$: a_1 and a_2 are neighbors.

- $R = \{r_1, ...r_l, ...r_p\}$ represent relay nodes with $p = |R|$ and $l \in [1, p]$. R is partitioned into o subsets (same as the number of anchor nodes) such that $R = \{R_1 \cup ...R_k \cup ...R_o\}$ and $\{R_1 \cap ...R_k \cap ...R_o\} = \emptyset$. Every subset R_k holds a number of relay nodes, i.e. $|R_k| = q$. q can be equal for all R_k, but it can also be different. Suppose subset $R_k = \{r_1, ...r_\ell, ...r_q\}$ with $\ell \in [1, q]$. Then, it holds for any $1 < \ell \leq q$ that node pair $(r_\ell, r_{\ell-1})$ are neighbors and node pair (r_1, a_k) are neighbors.

The flight recorder is not included in G since it does not have networking capabilities and a known relative position to other nodes. Depending on the location of the flight recorder, several nodes in G might be able to capture the emitted signals. G is mapped to a three-dimensional euclidean space, as illustrated in Figure 3.1. The three-dimensional x, y, z position of a node i is denoted as $\mathcal{P}(i) = (x, y, z)$ where $x, y, z \in \mathbb{R}$. Each node i is moving over time following a non-uniform trajectory.
When making a snapshot of the Eucledian space, each anchor node $a_k \in A$ is stationed at a known position $\mathcal{P}(a_k) = (x, y, 0)$ with $x, y \in \mathbb{Z}$. Each anchor node a_k has a set of $R_k \subseteq R$ relay nodes associated to it, together forming a *relay link*. Every relay node $r_\ell \in R_k$ has

an estimated position relative to its anchor, being $\mathcal{P}(r_\ell) = (\mathcal{P}(a_k).x + \epsilon_x, \mathcal{P}(a_k).y + \epsilon_y, z)$ with ϵ_x, ϵ_y indicating a divergence in location due to different ocean currents underwater. Every r_ℓ is physically attached to its neighbors which limits the horizontal displacement of r_ℓ to neighboring nodes, but it cannot be assumed that $\epsilon_x, \epsilon_y = 0$. Note that the physical link is not able to carry data signals; communication will be wireless.

The black box b is assumed to be at an unknown position $\mathcal{P}(b) = (x, y, z)$ with $z < 0$, emitting a periodic isotropic signal with a transmission time of 10 ms at an interval of 1 s at frequency bands between 36.5-38.5 kHz. The trajectory of b follows an unknown downward stream at an unknown velocity (i.e. b is sinking).

Onboard equipment

Anchor nodes are assumed to have a GPS sensor (for self-positioning), satellite communication equipment (to inform the necessary agencies of the flight recorder location), a motor (to control its position), an RF modem (to communicate with other anchor nodes), a hydrophone (for flight recorder detection) and an acoustic modem (to communicate with relay nodes in its relay link). Anchor nodes are equipped with a 12V 10AH battery (comparable to a battery in a scooter) and it cannot be recharged. Even though anchor nodes could theoretically recharge their batteries by the use of for instance solar power, battery recharging is considered out of scope for this study. Relay nodes are assumed to have a depth sensor and acceleration sensors (for self-positioning), a piezoelectric hydrophone (for flight recorder detection) and an acoustic modem (for communicating with other nodes underwater). All relay nodes are equipped with a 12V 500 mAH battery (comparable to a battery in a RC car). Batteries are assumed to be fully charged upon deployment but cannot be recharged while operating. From all anchor nodes in A, there is one anchor node that is elected as *head anchor*. This node is responsible for sending commands to other nodes and calculate the estimated location of the flight recorder. Hence, all other nodes forward their obtained measurements to the head anchor for processing. It is possible to alternate the head anchor role amongst all nodes in A to for instance spread out the energy consumption and communication load amongst all nodes, but this does not have to be the case.

The difference in depth between nodes in a relay link is here assumed to be constant. The recommended difference in depth is based on the depth the network must register ULB signals from, the communication range of an acoustic modem and a trade-off between localization delay and network reliability.

Suppose relay link $R_{link} = a \cup \{r_1, ...r_\ell, ...r_q\}$ consisting of anchor node a and q associated relay nodes where $1 < \ell \leq q$. Denote d as the vertical distance between a pair of consecutive nodes in R_{link}. One could choose d to be slightly less than the range of the acoustic modem to minimize the number of relay nodes in a link. However, if a relay node would malfunction (due to for instance the impact of the flight crash), the network could no longer localize on depths larger than the depth of the malfunctioning node. To improve network reliability, it is good practice to choose d as half the range of the acoustic modem, such that in case of a malfunctioning relay node r_ℓ, node $r_{\ell-1}$ will also be able to receive the signal such that the network can continue operating. This comes at the cost of a larger localization delay because the average number of hops on a data packet that needs to be forwarded to the head anchor increases. In case of no malfunctions, the transmitter power could be reduced to save battery since it is not required to utilize the full range of the communication link.

There exists a number of commercially available acoustic modems, with maximum operating ranges varying from 1000m to 6000m [10]. For this study, an acoustic link is assumed to have a maximum operating range of 3500m (see Appendix C) and d is initially chosen to be 1250m. To cover an ocean depth of 6000 meters, 4 relay nodes per relay link are deployed in the network. It is here assumed that a flight recorder ping transmitted from a depth of 6000 meters can still be detected at a depth of 5000 meters if the total distance between a receiver and the flight recorder does not increase over 2000 meters. Given that the sensor network was deployed near the crash site, this is considered a reasonable assumption. The performance of this configuration and alternative configurations are discussed in section 5.

3.2 Localization

From the localization techniques discussed in section 2.5, a TDoA-based approach is the best solution to the flight recorder localization case. Since there is no data communication with the flight recorder itself, there cannot be relied on information such as the time of transmission. Therefore, the ToA approach is not suitable. And since localization must be supported up to 6 kilometers, large distances between transmitter and receiver can be expected. Hence, AoA is not suitable due to its loss of precision over large distances. The TDoA approach assumes synchronization among receivers. This can be achieved by synchronizing receiver clocks before deploying in the ocean. In this work it is assumed that clock offset and clock skew are negligible for the network's predefined lifetime of 4 hours, hence all nodes in the network are assumed to be perfectly synchronized.
After a synchronized start, all relay nodes will start listening to signals in the frequency bands 36.5-38.5 kHz. All obtained data will be forwarded to the head anchor to calculate the location of the source of the pings. The overall localization process consists of 3 major steps:

- **Data collection**
  This is a synchronized process that runs on a relay node that describes what steps are involved from receiving an acoustic ping signal to forwarding measurement data to a head anchor.

- **Harvest orchestration**
  This is a process running on the head anchor that processes the arrival of measurements from relay nodes, groups them and decides when to calculate a location based on this data. It can also decide not to calculate a location at all, due to for instance inaccurate measurements.

- **Location estimator**
  This is the process of estimating a three-dimensional location of a source from the obtained hydrophone measurements.

These steps are discussed in more detail in the following subsections.

3.2.1 Data collection

A data collection strategy is responsible for enabling or disabling a hydrophone on a node, thus controlling its power consumption and detection capability, and processing signals

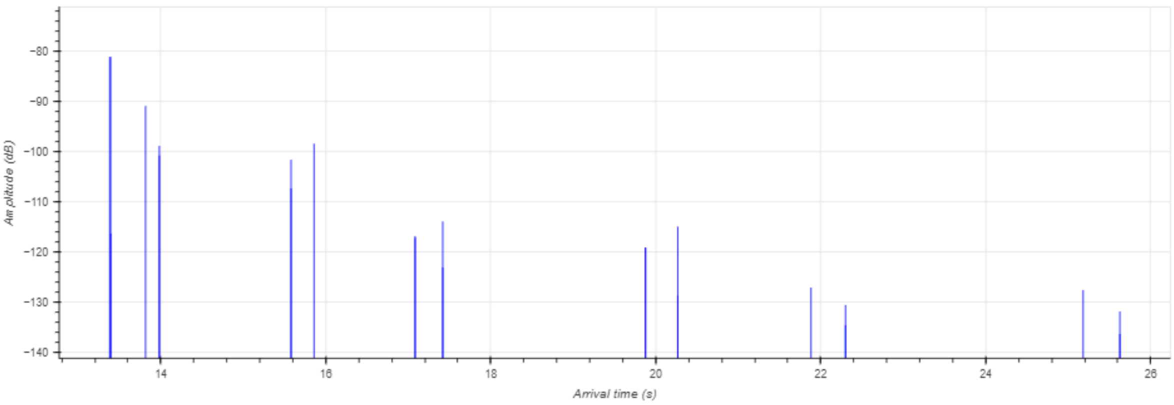
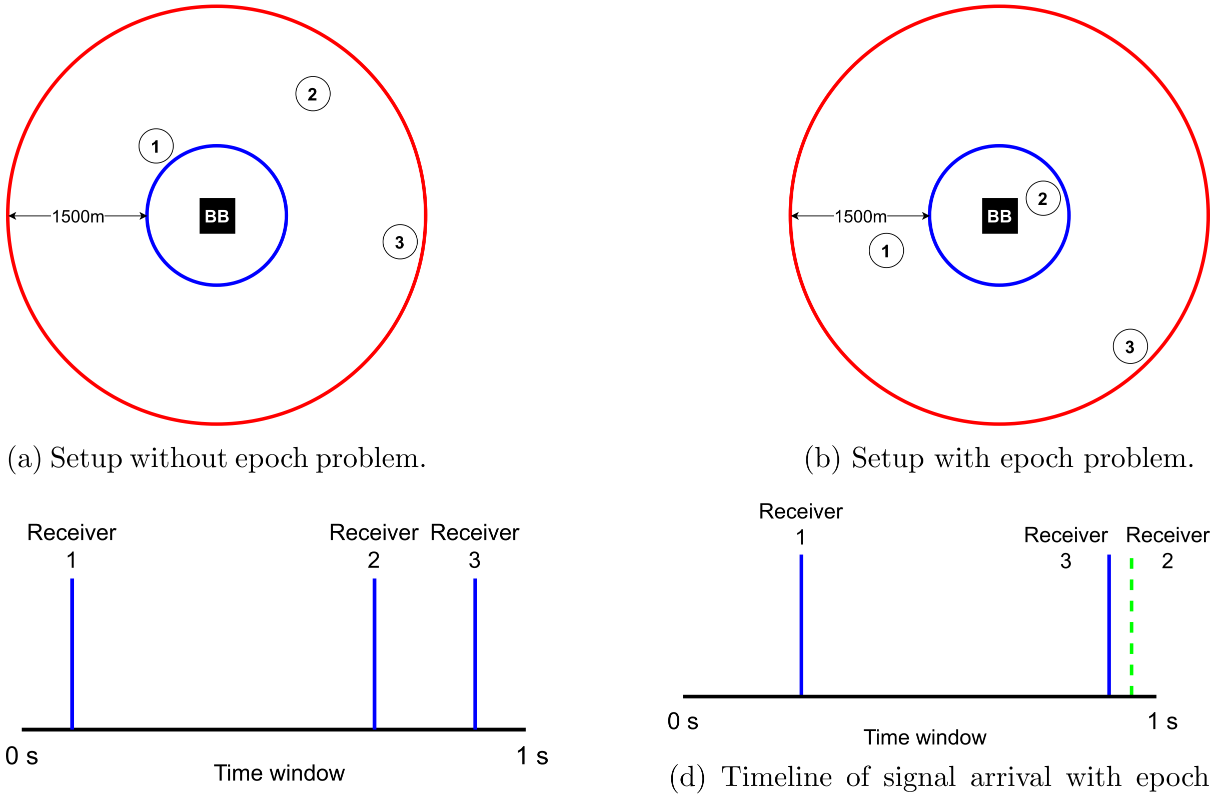


Figure 3.2: Arrival time-series with amplitude of the sender-receiver setup in Figure 2.2.

captured by the hydrophone. Starting a data collection strategy will go synchronized amongst all nodes in the network since timing is of high importance. Once started it follows the procedure described in Appendix A.1 which is called the *FindHighestPeak* data collection strategy. Once enabled, a hydrophone will listen for signals matching the characteristics of a ULB ping and starts a stopwatch so it can label incoming signals to a timestamp. It will register incoming signals for as long as the ULB transmission period. The time-series in which a hydrophone registers incoming ULB signals is referred to as the *time window* of detected signals. When the transmission period elapses, a new time window starts registering signals but these captured signals will be labelled to the next ULB transmission. Due to the multipath effect as described in section 2.2.1, during a single time window the same signal could be detected by the receiver multiple times on different moments. Since the location estimator (see 3.2.3) calculates possible source locations based on line-of-sight signal paths, the objective here is to filter out rays that have encountered a surface or bottom bounce. Since the receiver has no knowledge about the transmission, it cannot know solely on the timestamp of a detection if the received signal belongs to a new transmission, an already-detected transmission or even from transmissions a couple of time windows earlier.
To tackle this issue, the amplitude of the received signal is used to filter out reflected incoming signals. An example of a time-series of arrivals on a receiver is illustrated in Figure 3.2. Naturally, the first-received signal of a transmission is always the loudest. A reflected signal is always subject to damps and can therefore be distinguished from direct arrivals. Hence, the strategy finds the timestamp of the signal with the largest amplitude in the time window. This timestamp is considered the arrival of the ULB signal for this receiver and it is forwarded to the head anchor for location estimation.

Epoch problem

When creating time windows equal to the ULB transmission period, if there are no obstructions each hydrophone within range is guaranteed to pick up one direct line-of-sight signal of the ULB in each time window, due to the fact that the ULB transmissions are periodic and its period is fixed. This does not mean that every hydrophone picks up a signal from the same ULB transmission as the other hydrophones in that same time window. As illustrated in Figure 3.3, depending on the position of a set of receivers relative to the ULB, ambiguity to the received ULB transmission can be introduced. This issue is referred to in this work as the *epoch problem.* Knowing that the localization



(a) Setup without epoch problem.

(b) Setup with epoch problem.

(c) Timeline of signal arrival without epoch problem.

(d) Timeline of signal arrival with epoch problem. The green signal is yet to be transmitted in the snapshot in Figure 3.3b.

Figure 3.3: Epoch problem illustrated in a snapshot of when a set of receivers collectively starts listening to ULB signals. The colored rings show the position of propagating sound rays of the ULB at that moment. Each color indicates a different ULB ping. Note that relative receiver displacements are the same for Figure 3.3a and 3.3b; only their position relative to the black box is different. Assuming a sound speed of 1500 m/s and a ULB transmission period of 1 s.

method utilizes the TDoA of a signal amongst a group of receivers, this ambiguity may lead to a wrong source location estimate. For instance, consider the situation in Figure 3.3d. A TDoA algorithm is unable to distinguish different ULB transmissions (e.g. it cannot see the color in the figure). Therefore, it draws the conclusion based on the TDoA readings that the distance between the black box and receiver 2 is similar to the distance between the black box and receiver 3. However, the actual positions of receiver 2 and receiver 3 contradict this conclusion hence this will lead to a false black box location estimate. To minimize the probability of introducing the epoch problem, the distance between receivers must be minimized.[1] However, this would significantly reduce the detection range of the network. To still enable a large detection range in the network, a technique exploiting the periodicity of the ULB signal and slow displacement of nodes within time windows is applied in the location estimator algorithm described in section 3.2.3.

[1] The probability of introducing the epoch problem is minimized, but it can never be completely eliminated.

3.2.2 Harvest orchestration

A harvest orchestration strategy runs on a head anchor and is responsible for processing, validating and grouping measurements that arrive from hydrophone-equipped devices. Collected data consists of measurements from a synchronized observation window from various relay nodes. Therefore, in this data set the epoch problem might be present but that is not observable by the head anchor solely on the content of the measurements. When all data has arrived for a transmission, the harvest orchestration strategy will pass on all data belonging to that transmission to the location estimator. Also, the harvest orchestration strategy is responsible for commanding nodes to start or stop harvesting data. The strategy implemented for this study follows the procedure described in Appendix A.2 which is called the *AllNodesTimeout* harvest orchestrator strategy. Since hydrophone measurements do not necessarily arrive on the head anchor in chronological order, the AllNodesTimeout strategy will group measurements by measurement ID. The strategy is aware of how many hydrophone-enabled devices are in the network and the goal is to start localization with as many measurements as possible since that will improve localization accuracy. However, if a hydrophone misses a signal or a captured measurement will never reach the head anchor, it should not wait forever to start localization. Therefore this strategy uses a timeout. When the first measurement of a transmission arrives, a timeout timer is started to count down. After a certain amount of time, depending on the length of the relay link, all buffered measurements of the transmission will be forwarded to the localization estimator provided that the number of captured measurements matches the required minimum amount. If a measurement from all hydrophone-enabled devices reaches the head anchor earlier than the timeout timer elapses, the localization process starts earlier.

3.2.3 Location estimator

This strategy computes a three-dimensional location of a source utilizing the TDoA of captured signals and its receivers' coordinates. This study implements a widely-used approach in literature where the unknown values (the x, y, z coordinates of the source) are estimated by computing a best-fit solution to a system of equations.[18][17]
However, due to the epoch problem as described in section 3.2.1, it can be the case that some of the measurements contain signal peak timestamps belonging to a preceding ULB transmission. When solely estimating a location on the original set of measurements, the determined source location can have a significant error. Therefore, it is crucial to correct time-offsets introduced by the epoch problem. The process from obtaining a single set of measurements from a set of hydrophones to estimating a best-fit location of the ULB consists of three major steps.

Step 1: create permutations of time-offsets in measurements

Let there be n hydrophones where $1 \leq h \leq n$. Denote T_h as the peak-time measurement of hydrophone h. Denote M as the set of all T_h. Each T_h can have several offsets to its original peak-time. Denote I as the set of possible offsets. To denote measurements subject to a certain time offset $i \in I$, the following equation is used:

$$T_h(i) = T_h + i \times P \tag{3.1}$$

where P is the ULB transmission period in seconds. To get a set of every possible combination of measurement offsets, the n-ary Cartesian power of I is computed (referred to as $\mathcal{I}$) with cardinality $(\#I)^n$. For example, when $n = 3$ and $I = \{0, 1\}$, $\mathcal{I} = \{(0,0,0),(0,0,1),(0,1,0),(1,0,0),(0,1,1),(1,1,0),(1,0,1),(1,1,1)\}$. Each tuple $\mathcal{I}(\iota)$ where $\iota \in \{1, ...(\#I)^n\})$ denotes the offsets used in a set of hydrophone measurements M. Then, $(\#I)^n$ measurement sets are created, each M_ι having a different combination of possible measurement offsets following tuple $\mathcal{I}(\iota)$. For example, consider a measurement set $M = \{0.5, 1.0, 1.3\}$ and $P = 1$. Then, for a permutation ι where $\mathcal{I}(\iota) = (0, 1, 0)$, the offset-corrected measurement set becomes $M_\iota = \{0.5, 2.0, 1.3\}$. For each possible measurement set M_ι, a source location is calculated in step 2.

Applying this approach solves the epoch problem, if and only if the following condition is satisfied for every pair of hydrophones a, b with coordinates (x_a, y_a, z_a) and (x_b, y_b, z_b) respectively:

$$\frac{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}}{\#I - 1} < c \times P \tag{3.2}$$

where c is the propagation speed of sound, P is the ULB transmission period and $\#I$ is the number of different offsets used to make permutations. If equation 3.2 cannot be satisfied, this means that for a set of measurements there may exist hydrophone measurements that are strictly more than 1 epoch apart from one or more other hydrophone measurements in the set. Therefore, it may be the case that none of the permutations represents the actual positions of all hydrophones in the measurement set. However, equation 3.2 does not take into account the maximum range of the ULB. Suppose that the ULB has a detection range of at most 3 km, $c = 1500$ m/s and $P = 1$ s, an offset size of 3 will suffice because the nodes that are far away from the ULB and would introduce this problem are also too far away from the ULB to register any incoming ping. With this offset size, there are at most 3^n distinct sets of measurements to localize for.

Step 2: estimate source location of each permutation

For every measurement set created in the previous step, a location estimation is made during this step. Consider one measurement set consisting of n measurements (one measurement for each hydrophone). Denote (x_h, y_h, z_h) as the coordinates of hydrophone h. There is one source s that can send an acoustic signal. Denote the unknown coordinates of s as (x, y, z). Function T_h denotes the propagation time (s) from s to h. Let c be a constant speed of sound (m/s). Let $R_h = c \times T_h$ denote the distance (m) between s and h. Let $\tau_h = T_h - T_1$ denote the difference in propagation time between hydrophone h and hydrophone 1.
Note that

$$c\tau_h = cT_h - cT_1 = R_h - R_1 \tag{3.3}$$

Also note that we can substitute R_h for

$$R_h = \sqrt{(x_h - x)^2 + (y_h - y)^2 + (z_h - z)^2} \tag{3.4}$$

After some mathematical derivations on 3.3 and 3.4 by [18] we can get an equation formed as $0 = D_h + A_h x + B_h y + C_h z$ for hydrophones $3 \leq h \leq n$ where:

$$A_h = \frac{1}{c\tau_h} \times (-2x_1 + 2x_h) - \frac{1}{c\tau_2} \times (2x_2 - 2x_1) \tag{3.5}$$

$$B_h = \frac{1}{c\tau_h} \times (-2y_1 + 2y_h) - \frac{1}{c\tau_2} \times (2y_2 - 2y_1) \tag{3.6}$$

$$C_h = \frac{1}{c\tau_h} \times (-2z_1 + 2z_h) - \frac{1}{c\tau_2} \times (2z_2 - 2z_1) \tag{3.7}$$

$$D_h = c\tau_h - c\tau_2 + \frac{1}{c\tau_h} \times (x_1^2 + y_1^2 + z_1^2 - x_h^2 - y_h^2 - z_h^2) - \frac{1}{c\tau_2} \times (x_1^2 + y_1^2 + z_1^2 - x_2^2 - y_2^2 - z_2^2) \tag{3.8}$$

Suppose that equations A_h, B_h, C_h are combined in matrix M, then this equation can be rewritten in matrix form to get:

$$\begin{bmatrix} A_3 & B_3 & C_3 \\ \vdots & \vdots & \vdots \\ A_n & B_n & C_n \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = - \begin{bmatrix} D_3 \\ \vdots \\ D_n \end{bmatrix} \tag{3.9}$$

When having more than 5 measurements in the measurement set, one can observe that this becomes an overdetermined system since there are more equations than unknowns, therefore there generally exists no solution. Hence one must calculate the solution that minimizes the error. This is achieved by computing the Moore-Penrose pseudoinverse to M by using the singular value decomposition, resulting in matrix M^{-1}. Then one multiplies M^{-1} to the $-D$ vector on the right hand side, resulting in vector $\mathcal{P}$ that contains the minimized mean square error solution to (x, y, z) of source s. There only exists a solution when $n \geq 5$, hence 5 hydrophone measurements are required to estimate a source location using this method.

After finding a solution, its residual-vector R is calculated as follows:

$$R = M \cdot \mathcal{P} + D \tag{3.10}$$

Then the total sum of squares on R is calculated. This residual sum of squares is placed on a buffer together with solution $\mathcal{P}$.

Step 3: find solution with least residual sum of squares

Suppose that there exists a measurement set M_{correct} where every hydrophone measurement truly relates to the same ULB transmission. In this case, there is no epoch problem. If localization on measurements from M_{correct} results in a low source position error, it is believed from empirical evidence that this solution also has a low residual sum of squares. While satisfying equation 3.2, there exists a $\iota \in \{1, ..., (\#I)^h\}$ such that $M_\iota = M_{\text{correct}}$. Hence, from all solutions calculated in step 2, the solution with the least residual sum squares is forwarded as the estimated source location.

Determining the error on the estimated location

To assess the accuracy of the estimated location, the eucledian distance error ϵ can be calculated between estimated coordinates $(\hat{x}, \hat{y}, \hat{z})$ and actual coordinates (x, y, z):

$$\epsilon = \sqrt{(\hat{x} - x)^2 + (\hat{y} - y)^2 + (\hat{z} - z)^2} \tag{3.11}$$

Solving for $\hat{x}, \hat{y}$ and $\hat{z}$ by using this method assumes a constant sound speed and a straight-line propagation model. For location estimations over larger distances, ϵ may be larger as opposed to shorter distances due to these assumptions. How the choice of sound speed impacts the localization error is assessed in section 5.2. The proposed number of permutations of 3^h measurement sets is quite demanding for a small processor running on an anchor node. Therefore, there are a number of suggestions to apply here:

- To limit the number of permutations, drop permutations of offsets for which you can know in advance are not possible. For instance, if the distance between a pair of hydrophones h_1 and h_2 is small, there is no need to calculate permutations where it assumes that there are more than 2 epochs between h_1 and h_2.
- In the beginning of deployment it is known that not all relay nodes are deployed or far away from their anchor. Hence, the offset size may begin with a size of 2, and when the distance between nodes becomes larger the offset size may be increased.
- Previously determined permutations may be remembered for future location estimations. If the ULB sends a ping on a short interval, it is likely that the nodes remain in the same epoch relative to the new location of the black box. If the residual sum of squares on a more recent measurement set using offsets from the previous localization is similar to (or better than) the residual sum of squares of the previously calculated best-fit, there is no need to recalculate all permutations.
- Instead of calculating the location of all measurement set permutations on the head anchor, the head anchor may forward all measurements to an off-site powerful processor that performs the location estimation instead via a satellite link.

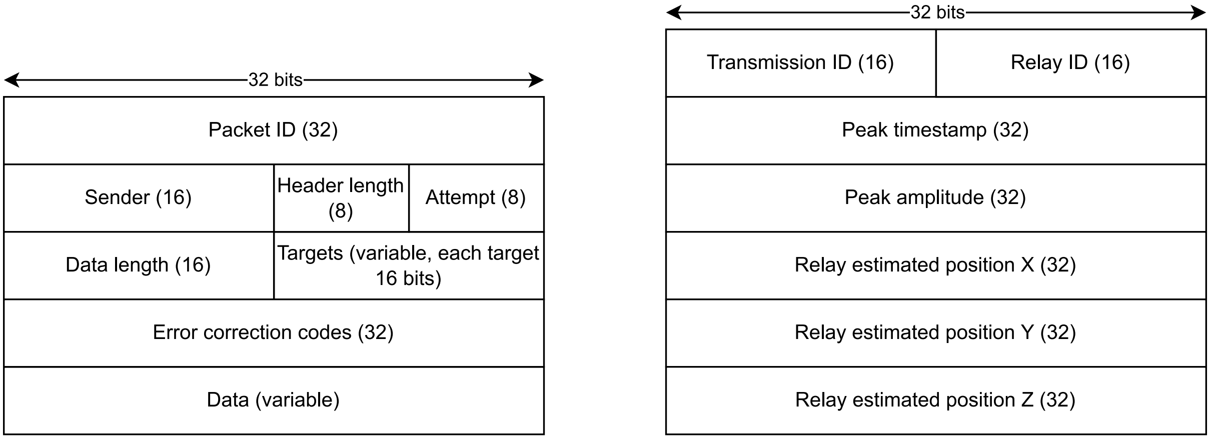
3.3 Communication

As discussed in section 2.2, the best way to communicate underwater over large distances ($>$ 100m) is by means of acoustics. Therefore, all relay nodes communicate using acoustic modems. The anchor nodes have the ability to use the air as propagation medium, so inter-anchor communication is achieved by means of radio-frequency modems. Both approaches are discussed in this section, with emphasis on underwater communication since that covers the most important part of this study.

3.3.1 Underwater acoustic communication

After processing a ULB measurement, a relay node must forward it to the head anchor for flight recorder location estimation. Since the structure of the network is known by the sender, a simple hop-by-hop scheme can be used for routing a message to an anchor node. The relay node will send the measurement to the first node upwards in the relay link relative to itself. Any relay node in the relay link will store and forward incoming measurements from nodes lower in the link to the next node. Eventually the measurement will be received by the anchor node on top of the relay link.

The frequency range of the acoustic modem is divided into channels, each channel having an equal channel bandwidth. Every relay link in the network has exclusive access to



Packet ID (32)		
Sender (16)	Header length (8)	Attempt (8)
Data length (16)	Targets (variable, each target 16 bits)	
Error correction codes (32)		
Data (variable)		

(a) Header

Transmission ID (16)	Relay ID (16)
Peak timestamp (32)	
Peak amplitude (32)	
Relay estimated position X (32)	
Relay estimated position Y (32)	
Relay estimated position Z (32)	

(b) Data: measurement sent by relay node

Figure 3.4: Structure of a data packet for acoustic transmission

one channel, therefore it is assumed that underwater signals coming from a relay link do not interfere with signals from other relay links. For this study, an acoustic modem is equipped in each node with specifications based on commercially available acoustic modems [10]. The specifications of this modem can be found in Appendix C. Assuming that there are 3 relay links in the network, the available frequency band of $18 - 34$ kHz is divided in three non-interfering channels each having a bandwidth of 5kHz.

Packeting measurements

To send a measurement to another node, it must be translated to data packets that can be sent over the channel. Figure 3.4 shows the structure of a packet header and the structure of the data of a measurement from a relay node. This packet structure supports multicasting, which could be useful in networks where a signal could reach multiple nodes in the relay link such that unnecessary hops can be skipped for packets that are meant to be processed by multiple nodes in the link (i.e. commands from an anchor node). Also the header structure supports inclusion of error correction codes and a variable payload size. Each measurement has a fixed size of 320 bits and contains everything the location estimation algorithm needs to know from a node for its calculations. Each measurement is transmitted via a separate packet. In case of a packet drop, only one measurement needs to be retransmitted without the risk of other measurements being discarded by the harvest orchestrator due to extra communication delay.

Forward error correction

The error correction codes included in a packet header are based the Reed-Solomon error correction algorithm (see section 2.4.4). The number of error correction codes (ECCs) is fixed on 4 bytes. According to equation 2.12, this means that up to 2 erroneous bytes could be corrected. Note that erasures are not considered since the receiver does not know the location of errors in advance. But on top of error correction, it serves also as error detection (checksum) on the receiver side. If the number of ECCs change, the packet header structure changes accordingly. The error correction scheme here is meant to prevent retransmissions in scenarios with little noise on the channel where only a few bits would flip. In cases where the SNR is too large, FEC is of no use. Increasing the number of ECCs also means that the packet size would increase, hence the probability of error would also increase. For noisy channels, it is better to reduce the symbol alphabet size in the constellation scheme rather than increasing the number of ECCs.

Packet acknowledgment and retransmissions

To ensure reliable data transfer, each packet sent must be acknowledged by the receiver (except for acknowledgment packets). Every packet includes a unique ID in the header such it can be matched with acknowledgments. The packet acknowledgment and retransmission process is described in Appendix B.1. Every packet that must be acknowledged is stored in a buffer along with its recipients. Upon arrival of an acknowledgment by one of the original recipients, the recipient will be removed from the buffer. If all recipients of the packet have sent an acknowledgment, the packet is also removed from the buffer. An acknowledgment timeout is in place. If at the start of a transmission slot certain packets crossed the acknowledgment deadline, these packets will be removed from the buffer and are retransmitted to the recipients that did not yet acknowledge. However, if a packet has reached its maximum transmission attempts, it will not be transmitted again and this packet is considered to be dropped. In practice this means that the measurement will not arrive at the head anchor, possibly leading to a lower localization accuracy. While processing the transmission queue of a transmitter, acknowledgment packets have priority over other types of packets.

To minimize acknowledgment traffic, acknowledgment packets are grouped by recipient and all acknowledgments for that recipient are wrapped in a single packet. Each recipient in the acknowledgment packet payload is considered to be 32 bits. The reason to merge acknowledgments to a single packet is to minimize overhead introduced by packet headers. After the payload of an acknowledgment packet is crafted, the *data length* attribute in the packet header is set accordingly.

Transmission slot scheduling

To prevent signal collisions as mentioned in section 2.4.6, a transmission schedule tailored to this approach must be created. Due to the long propagation delay underwater, it is desirable that each individual node is able to calculate the schedule by itself, hence not relying on information from another node in order to determine the schedule. It is chosen to use a schedule where the length of a time slot for a node remains the same throughout time windows because the node can calculate the next transmission time slot independently of other nodes. Due to the structure of relay links, transmission slot order should be from bottom-to-top in terms of node position. When doing so, the number of queued packets for transmission will increase for relay nodes higher in the relay link since they forward messages received from relay nodes below them. Hence it makes sense to adjust the slot time accordingly. A node is not allowed to transmit outside of its transmission slot. If a packet was being transmitted while a transmission slot elapses, the packet is considered lost and will not be removed from the transmission queue. Hence the packet will be sent again next transmission slot. Due to space-time uncertainty (section 2.4.6) and the bottom-to-top slot order, it is important that after a time slot finishes, there is some time before the next transmission starts to prevent TX-RX collision and possibly RX-RX interference. This idle time between time slots is called the *guard time* and is calculated as follows:

$$g = \frac{d}{c_{min}} \tag{3.12}$$

where d is the inter-relay distance and c_{min} is the minimum propagation speed of sound underwater. Based on the Munk profile, $c_{min} = 1492$ m/s. Hence for an inter-relay distance of $d = 1250$ m, the guard time is 0.84 s. This is a worst-case estimation for situations where nodes reside in a sound speed profile of 1492 m/s and a transmitter fully uses its time slot to send packets. But since nodes do not inform other nodes on their transmission queue length, it is fail-safe to assume this worst-case scenario.
Suppose relay link k with $|R_k|$ relay nodes in the link. Suppose node $n \in \{0, 1, ..., |R_k|\}$ where $n = 0$ is the anchor node on top of the relay link and $n = |R_k|$ is the relay node deepest in the link. When knowing the guard time and the number of relay nodes in a relay link, one can calculate the total window time for this time slot approach:

$$W = 0.5 \cdot m(|R_k|^2 + |R_k|) + g \cdot |R_k| + 0.5m \tag{3.13}$$

where $|R_k|$ is the number of relay nodes in a relay link and m is the time slot multiplier. m can be used to extend transmission slots such that the guard time has less overhead on the total window time. However, larger transmission slots come at the cost of larger communication delay. Note that it is important to take the maximum distance of the acoustic modem into account. Equation 3.13 assumes that acoustic signals from the anchor node cannot reach the next node that must receive signals from the node below it ($n = |R_k| - 1$). In the case of $|R_k| = 4$ with $d = 1250$ and an acoustic modem range of 3500 meters, this holds since the distance between those nodes is 3750 meters. Therefore, no guard time is required after the time slot of the anchor node has elapsed. If the nodes do interfere, one must increase the guard time after the time slot of the anchor node to avoid interference, leading to an alternative total window time:

$$W_{alt} = 0.5 \cdot m(|R_k|^2 + |R_k|) + g \cdot |R_k| + 0.5m + max(g \cdot (|R_k| - 2), g) \tag{3.14}$$

One can also calculate the time slot length w for a node n:

$$w(n) = \begin{cases} 0.5m & \text{if } n = 0 \\ m & \text{if } n = |R_k| \\ m + w(n+1) & \text{if } 0 < n < |R_k| \end{cases} \tag{3.15}$$

The anchor node ($n = 0$) will not send measurement packets, but only acknowledgments of received packets. Therefore it has a shorter transmission slot than other nodes. Applying equation 3.15 with a network consisting of 4 relay nodes per relay link and a multiplier of 1, the slot schedule as illustrated in Table 3.1 can be formed.

Relay link node	**Time slot duration (s)**
Node 0 (anchor)	0.5
Node 1	4
Node 2	3
Node 3	2
Node 4 (bottom)	1

Table 3.1: Transmission time slot duration for $|R_k| = 4$ and $m = 1$. With $g = 0.84$, this schedule has a window time $W = 13.86$ s.

Required data throughput

For determining the required minimum data throughput, relay links with 4 relay nodes per link are assumed, where each relay node is vertically spaced by 1250 meters. The ULB sends a ping every second and all captured pings are supposed to be forwarded to the head anchor in the network. The communication schemes in section 3.3.1 are considered where each measurement is contained in exactly 1 packet. Since each relay link uses its own channel in the available frequency range, the required data throughput is considered for 1 relay link, but is the same for all other relay links in the network. For the minimum required throughput it is assumed that all transmissions arrive to its recipients on the first attempt. The minimum required throughput (bps) can be calculated with:

$$bps_{min} = max_{n=0}^{|R_k|} \frac{b_m(n) + b_{ack}(n)}{w(n)} \tag{3.16}$$

where $w(n)$ is the time slot duration for node n. $b_m(n)$ denotes the required bits for sending measurements from node n, which can be calculated as follows:

$$b_m(n) = \begin{cases} 0 & \text{if } n = 0 \\ 320 \cdot \lceil W/T \rceil & \text{if } n = |R_k| \\ 320 \cdot \lceil W/T \rceil + b_m(n+1) & \text{if } 0 < n < |R_k| \end{cases} \tag{3.17}$$

where W is the window time (equation 3.13) and T is the ULB transmission period. Since relay nodes forward their measurements to their anchor through a hop-by-hop mechanism, every relay node except the deepest relay node must forward both its own measurements and the measurements it has received from below. The anchor node does not have to forward measurements over the acoustic channel.
$b_{ack}(n)$ is the required bits for sending acknowledgments from node n, which can be calculated as follows:

$$b_{ack}(n) = \begin{cases} 0 & \text{if } n = |R_k| \\ 32 \cdot b_m(n+1) + 128 & \text{if } n < |R_k| \end{cases} \tag{3.18}$$

Every node must send an acknowledgment packet when it receives a measurement from another node in its relay link. The number of required acknowledgments therefore strongly depends on the number of received measurements. Only the deepest relay link does not have to send acknowledgments since it does not forward measurements from other nodes.

bps_{min} here depends on the window time W, which in turn depends on the guard time g and the slot time multiplier m. Suppose a guard time of 0.84 s, a ULB transmission period of 1 s and 4 relay nodes per relay link. Here, a larger slot multiplier effectively leads to a lower bps_{min} since there is less overhead from g. Figure 3.5 shows W and bps_{min} as a function of m. As can be seen in the figure, increasing the time slot duration does have a strong effect on lowering the minimum data throughput, but this effect weakens on larger multipliers. The window time increases linearly over time slot duration increasing. Hence, depending on the available bandwidth an appropriate time slot multiplier can be chosen.

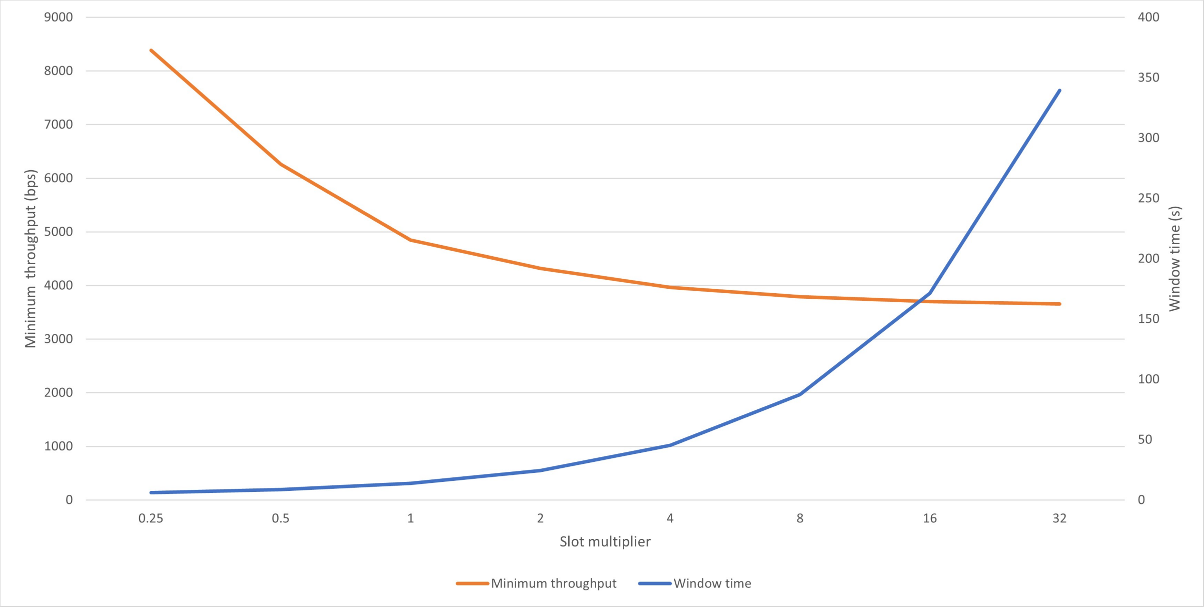


Figure 3.5: Minimum required data throughput (bits per second) and window time (seconds) versus time slot multiplier for $g = 0.84$, $T = 1$ and $|R_k| = 4$.

Signal-to-noise ratio estimation

The SNR is important to understand the implications of noise on the data that is being transferred. A lower SNR (more noise) may lead to more deviations in the arriving signal. That means that the channel capacity decreases. Also, the SNR can have impact on the choice of modulation symbol alphabet size. To estimate the SNR, the expected noise levels and transmission loss must be determined. By assuming a center frequency of 22kHz and an inter-relay distance of 1250 meters, the transmission loss can be calculated by using equations 2.3-2.5 coming to a total of 70.56 dB. For noise levels, a shipping activity of 0 is assumed since no ships are expected to be at the crash site upon deployment of this network. A wind speed of 12 m/s (wind speed of 6 bft) is assumed. Using equations 2.6-2.9, noise is estimated on 6.06dB. The Source Level (SL) is calculated based on a Transmission Voltage Response of 140 dB re $\mu Pa/1V$ @1m and a power supply of 12 V, leading to an effective SL of $140 + 10\log_{10}(12) = 150.79$ dB. According to equation 2.10, the SNR equals 74.17 dB ($2.612 \cdot 10^7$ Watts). Shannon's theorem (equation 2.14) estimates on a 5 kHz bandwidth a maximum throughput of 61.5 kbps, but it is assuming a theoretical symbol size $m \approx 5100$ which is practically not feasible. A lower symbol size must be chosen while still accommodating the required data throughput. What symbol size is appropriate is discussed in the following paragraph.

Signal modulation

This approach utilizes the phase-shift keying constellation scheme. The whole network uses the same symbol alphabet size which can be selected based on the available throughput, the SNR and the required throughput. Since the SNR is quite good, not many re transmissions are expected. With 3 relay links in the network, a channel bandwidth of 5kHz can be achieved leading to a maximum throughput of 10 kbps assuming a noise-free channel and a symbol size of 2. The expected required throughput for a transmission schedule with a multiplier of 1 is 4.8 kbps. Therefore it is expected that a BPSK modulation scheme (symbol size of 2) will suffice for this application.

3.3.2 Terrestrial communication

Terrestrial communication involves communication above the sea water and relies on radio communication. Since this study focuses on underwater communication, terrestrial communication will be modeled much simpler. RF channels have a much larger bandwidth and the time-space uncertainty effect is negligible due to the large propagation speed of light ($3 \cdot 10^8$ m/s). As opposed to underwater acoustic communication, RF communication assumes that there is no signal collision due to some signal collision avoidance protocol included in the RF modem. Every transmitted data packet will arrive at the receiver.

In this study, anchor nodes utilize RF modems with specifications from [8]. Whenever an anchor node receives a message to forward to the head anchor or it needs to forward its own measurement, it will transmit it immediately.

Chapter 4

Underwater Acoustic Localization Simulator

To assess the approach discussed in Chapter 3, a simulator is built using Python 3.9, called the *Underwater Acoustic Localization Simulator* (UWALSim). This simulator supports both underwater acoustic communication and underwater sound propagation modelling. As of writing, there was no simulator available that supports both of these features. Having both features, the entire process from the transmission of a ULB ping until calculating a source location can be modelled and its results can be assessed. The simulator is built on top of the *SimPy* discrete simulation engine [29]. The simulator comes with configuration files, where all properties and energy profiles can be inserted. The simulator is designed to allow for easy extension to certain devices, strategies, communication equipment and propagation models. The overall structure of UWALSim is illustrated in Figure 4.1. Each of the 7 modules is discussed in this chapter, where design choices and relevant implementation-specific details are presented.

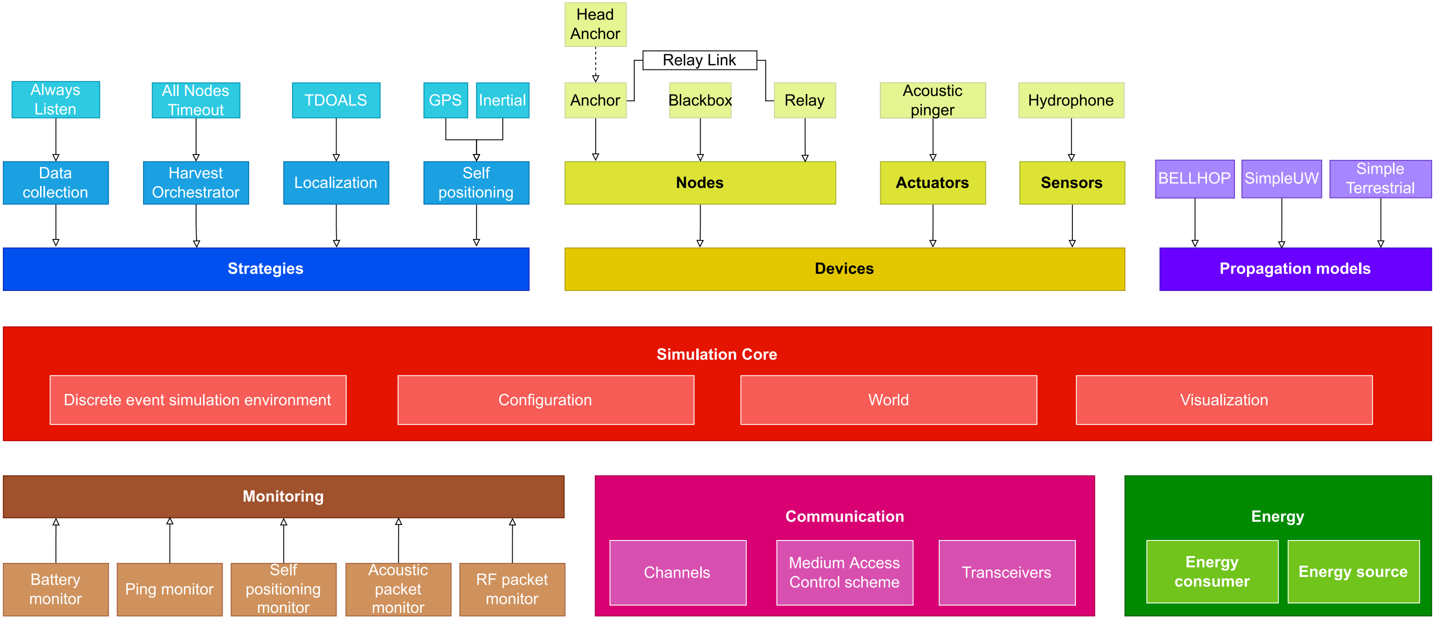


Figure 4.1: UWALSim module topology

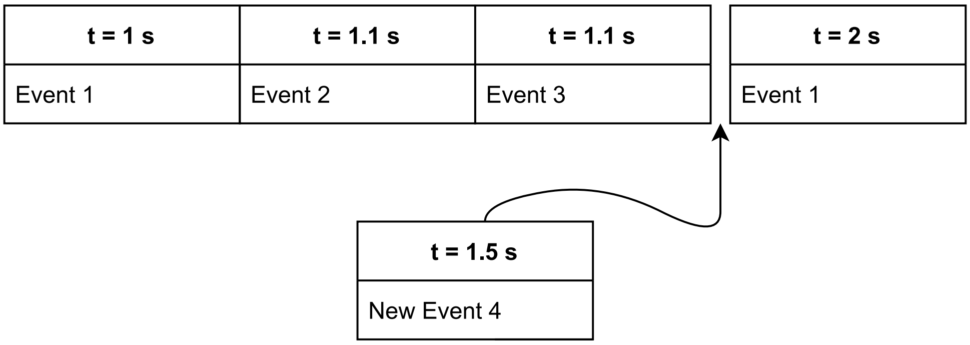


Figure 4.3: Example of the event queue and scheduling a new event.

4.1 Simulation core

The SimCore module is the heart of UWALSim, where the discrete simulator, environment, configuration and visualization is controlled. The discrete simulation engine is powered by SimPy [29]. Before starting a simulation, a SimPy simulation environment is created. All entities in UWALSim that require this SimPy environment can import the SimCore Python package to interact with the object. SimPy, and most other discrete simulator engines, work with a large queue called the *event queue*. As the name suggests, this queue contains events that need to be fired at some point in simulation time. The queue is sorted by ascending simulation time. When scheduling an event, it is placed in the queue by the simulation time the event is supposed to fire as illustrated in Figure 4.3. Events can be scheduled once or they can be scheduled repeatedly. When running the simulation, the engine processes the event queue as fast as possible. Each time it processes a new event from the queue, it updates its current simulation time to the value of the simulation time that fires the event. Events cannot be inserted directly in the queue given a desired simulation time to fire the event. Scheduling events can be achieved by relative time delay (i.e. event will start in 1 second from now) or they can be fired by other events. SimCore also reads the configuration files required for simulation and stores them in-memory for quick access during simulation. An example of these configuration files can be found in Appendix E.

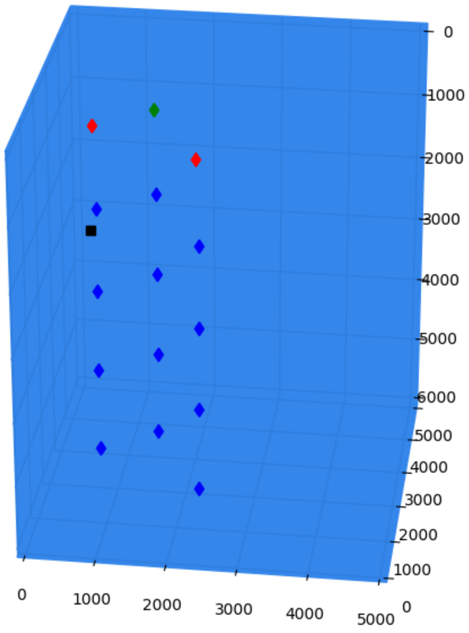


Figure 4.2: Example of visualizing nodes in an oceanic environment using UWALSim. Red dots represent anchor nodes, the green dot represents the head anchor, blue dots represent relay nodes and the black square represents the black box.

The world module defines the physical environment that the simulation takes place in. For the purpose of this study, the world consists of an ocean with specific properties (e.g. depth, bathymetry and altimetry). The world and all nodes residing in it can be visualized in a 3D model. When the simulator is running, it shows the displacement of nodes in the ocean. The user of the simulator can easily change perspective on the setup in the model. Figure 4.2 shows an example of how the environment is visualized in UWALSim. Prior to running the simulation, an expected trajectory for each node on several depth levels can be configured, representing displacements introduced by ocean currents and node buoyancy. During the simulation, node displacements will be calculated according to this provided trajectory.

4.2 Devices

UWALSim models three types of devices: nodes, actuators and sensors. Nodes are categorized in Anchor, Relay and BlackBox subtypes. There is also a RelayLink object that associates anchor nodes and relay nodes to each other. The RelayLink contains properties that indicate how these nodes are physically connected, which is used in node displacement calculations given some ocean current profile and buoyancy of the node. Upon initializing the simulator, nodes can be equipped with actuators and sensors and they can be placed in the ocean. The simulator expects one Head Anchor to be initialized. A Head Anchor is a subclass of Anchor and enables capabilities of initializing a harvest orchestrator and location estimation module. Also, the Head Anchor is the recipient of RF-packets sent by other anchors with measurement data.
Actuators are for instance a motor (for controlling the position of an anchor node) or an acoustic pinger. An acoustic pinger is installed in a BlackBox instance and has a configurable periodic ping profile. When starting the simulator, a timer process is started in the AcousticPinger object that repeatedly schedules events representing an acoustic ping transmission. Each time a ping transmission is triggered, the AcousticPinger object starts crafting a BELLHOP propagation model query and calculates noise and path loss profiles from equations 2.3-2.9. The simulator finds all Hydrophone sensor instances residing in the ocean and BELLHOP calculates signal arrival times for each hydrophone. Based on the provided signal strength of the ULB, the calculated sound power losses, BELLHOP's predicted amplitude of the arrived signal (taking into account possible bounces), the SONAR equation (equation 2.10) and the Detection Threshold (2.11), the simulator estimates for each hydrophone if it would have heard the ping. If it decides that a hydrophone h would have been able to observe the ping, it lets SimPy schedule an event on h to fire in simulation time $t + T_h$ where t is the current simulation time (upon triggering the acoustic ping transmission) and T_h is the propagation time calculated by BELLHOP from the black box to h. Note that BELLHOP arrival time series include multipath arrivals. Hence, for a single transmission there may be several scheduled arrival events for each hydrophone. It is up to the data collection strategy to deal with these multipath arrivals. After the Hydrophone event of detecting a black box ping fires, the simulator first checks if the hydrophone was turned on by the data collection strategy. If not, the simulator discards the detected measurement. If turned on, the simulator fires an event on the data collection strategy including details on the detected ping (such as the time of detection and the amplitude of the received signal).

4.3 Strategies

The strategies described in this study are implemented in one of the submodules of the strategies module. What strategy implementation to pick is set up before running the simulation.

4.3.1 Data collection

There are two implementations for data collection strategies: *AlwaysListenHighestPeak* (as described in section 3.2.1) and *AlwaysListenMagicSelection*. Their main difference lies in how they select a signal to be the alleged direct line-of-sight signal from the receiver to the flight recorder. After running a signal-arrival query on the chosen propagation model, the propagation model returns a time-series of signal arrivals on the receiver. The BELLHOP propagation model also includes an estimation on the signal amplitude for each arrival. The AlwaysListenHighestPeak strategy uses this amplitude to select the strongest signal to be the direct line-of-sight signal. The AlwaysListenMagicSelection strategy does not use the amplitude of a signal to determine the direct line-of-sight signal, but it uses simulation meta-data from the propagation model query to match the arrived signal to the transmission ID allocated by the AcousticPinger transmitter. Therefore, this MagicSelection strategy exactly knows to what transmission the received ping signal belongs. For any received ping belonging to a certain transmission ID, it only forwards the ping that it received first. Even though this strategy cannot be applied to real-life applications (since it uses simulation meta data), this strategy can be useful in situations where it is desirable to neglect multipath-arrivals and the epoch problem. This can be useful to measure the impact of other strategies or configurations on the localization error. Both strategies use a configurable processing time to simulate delay due to processing a series of measurements.

4.3.2 Harvest orchestrator

The implementation of the AllNodesTimeout harvest orchestrator follows the procedure as described in section 3.2.2. Timeouts are modeled by a separate SimPy process that is executed when the first measurement of a certain transmission arrives. The process is sleeping for the configured strategy timeout time (how much time the strategy should wait for new measurements after the first measurement has arrived). After this timeout elapses, all buffered measurements are forwarded to the localization strategy. The strategy knows how many hydrophones are in the network. If the number of buffered measurements reaches the number of hydrophones in the network, the timeout process is interrupted before the timeout time elapsed and starts forwarding its measurements immediately.

4.3.3 Localization

The TDOALS implementation follows the localization strategy as described in 3.2.3. The sound speed constant used in this approach can be set in the configuration file. For the list of h received measurements and a configurable offset size, it creates permutations of the measurement set according to the procedure described in section 3.2.3. This leads however to a large number of measurement sets. When using an offset list of $\{0, 1, 2\}$ the number of permutations is 3^h. When deploying 3 anchor nodes and 4 relay nodes per anchor, this could lead to at most 15 hydrophone measurements, hence $3^{15} = 14348907$ different sets of measurements.

To speed up the localization calculations in step 2 from section 3.2.3, the implementation of that step is completely decoupled such it can be run in parallel on multiple cores. All permutations are placed on an input buffer and p independent workers read from this input buffer and place the output (which are estimated x, y, z coordinates of the source

and its residual) on an output buffer. p represents the desired number of parallel processes and is configurable. The main process checks the output buffer and saves only the solutions with the lowest residual it has encountered so far. After all permutations of measurement sets have been processed by the workers, the solution with the lowest residual is forwarded to assess the accuracy of the solution. The simulator calculates the localization error by using equation 3.11. The location of the black box when it was transmitting the ping is used to assess the error of the estimation.
Even though the implementation exploits parallelism, running simulations with an offset size of 3 generates too many permutations to run on an average system without introducing long delays in obtaining simulation results. Therefore, the implementation remembers the offsets of the previously calculated best-fit solution. On a new incoming set of measurements, it first tries localization using the offsets previously used. Only if the residual sum of squares is much higher than the residual sum of squares of the previous best-fit solution, it will recalculate all possible permutations. It is expected that recalculations do not happen often, such that severe calculation delays can be limited.

4.3.4 Self-positioning

There exist two self-positioning strategies: GPS self-positioning (used by anchor nodes) and Inertial self-positioning (used by relay nodes). New position estimations are made every T seconds and come with a configurable energy consumption profile, depending on the strategy used.

GPS

This strategy uses an error that is added to the actual position of the node for each self-positioning query. The error in GPS coordinate estimations can be approximated by a normal distribution [34]. Mean values and standard deviations differ for longitude, latitude and altitude and are provided in the simulator configuration. Since the anchor node is floating on the ocean surface, it discards the estimated altitude provided by the GPS receiver and fixes its Z-coordinate to 0.

Inertial

This strategy models self-positioning based on node movement relative to its previously estimated location. Horizontal displacement is assumed to be registered by acceleration sensors, where vertical displacement is assumed to be registered by pressure sensors. The exact implementation of each of these sensors is out of scope for this study, only their expected measurements are modelled. When a relay node is dispatched from its anchor, the estimated GPS position of the anchor is given as initial position of the relay node. Since the new horizontal position of the node is based on the previously determined estimated position, this strategy must model a cumulative error for those position queries. The estimated $\hat{x}, \hat{y}$ coordinates are sampled as follows:

$$\begin{aligned} \hat{x} \sim& \mathcal{N}(\Delta_x, \sqrt{\Delta_x} \cdot 0.5\epsilon_x \cdot \frac{|\Delta_x|}{T}) \\ \hat{y} \sim& \mathcal{N}(\Delta_y, \sqrt{\Delta_y} \cdot 0.5\epsilon_y \cdot \frac{|\Delta_y|}{T}) \end{aligned} \tag{4.1}$$

where $\mathcal{N}(\mu, \sigma)$ is the Normal distribution with mean μ and standard deviation σ. Δ_θ is the displacement of coordinate θ in a time period of T seconds. ϵ_θ is the configurable error (m) of coordinate θ per meter. The horizontal position error of a node with this inertial estimation model increases over time. The inaccuracy of the sensor is modelled to be greater when the node travels faster.
The estimated $\hat{z}$ coordinate is sampled as follows:

$$\hat{z} \sim U(\Delta_z - \epsilon_z, \Delta_z + \epsilon_z) \tag{4.2}$$

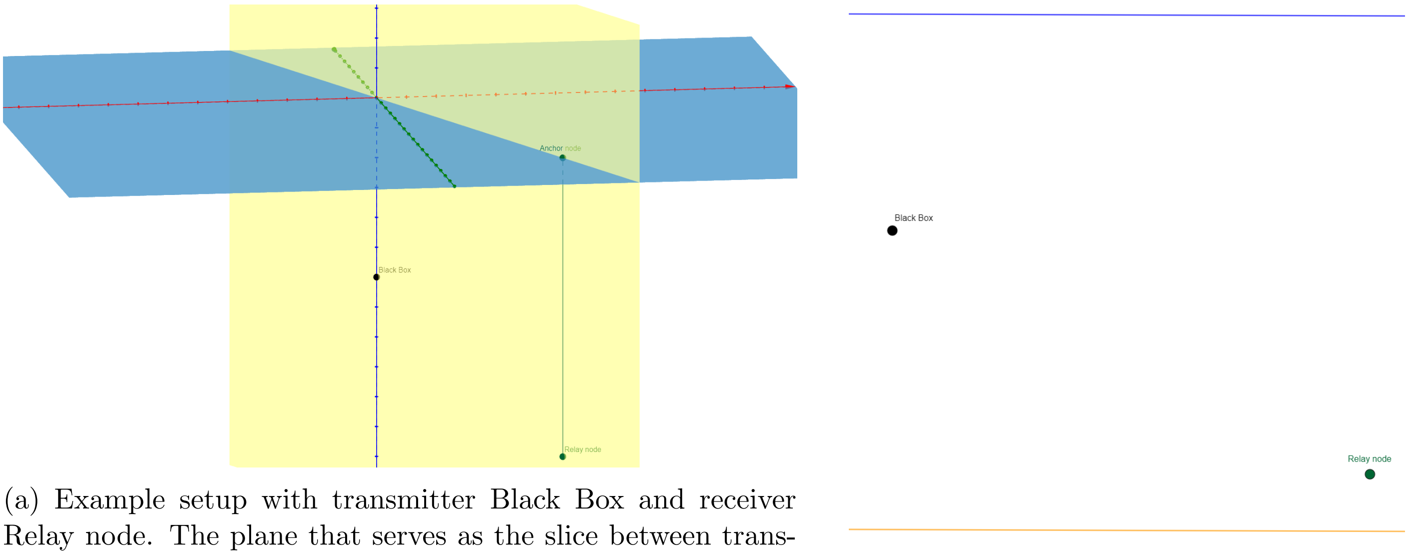
where $U(a, b)$ is the Uniform distribution between boundaries a and b where $a < b$. Δ_z is the displacement of coordinate z between the previous self-positioning query and the current self-positioning query and ϵ_z is the configurable maximum error (m) on the actual depth of the device.

4.4 Propagation models

A propagation model is a module that, given a source location, a signal frequency and a list of positions of possible receivers in the channel, calculates the arrival time(s) of the signal to each receiver. Optionally, it can model multipath arrivals and calculate extra features like the angle of arrival, received signal amplitude and the number of surface/bottom bounces. Propagation models are used for ULB signal propagation and communication modem signal propagation. There are three propagation models implemented in UWALSim: BELLHOP, SimpleUW and SimpleTerrestrial. Each of the propagation model implementations are discussed in this section.

4.4.1 BELLHOP

The BELLHOP propagation model implementation in UWALSim is an interface to the BELLHOP 2D beam tracing model (section 2.2.3), provided by the *ArlPy* Python library that aids in underwater acoustics, signal processing and communication modelling [4]. The executable of BELLHOP must therefore be in the system path of the computer running UWALSim. BELLHOP requires input files that describe the environment and ray tracing query. After BELLHOP executed the query, it writes back an output file with the query results. The BELLHOP interface provides the required files and reads back the file containing the result.
First, the underwater environment is defined. Specific underwater properties can be set through the simulator's configuration file, for instance the ocean depth, the altimetry profile and the bathymetry profile. BELLHOP supports a dynamic sound speed profile. UWALSim provides the Munk sound speed profile (equation 2.1) upon initialization of the environment. Since BELLHOP works with a two-dimensional environment and UWALSim works with a three-dimensional environment, a conversion must be made to make them compatible. Upon triggering a ray-trace query on a transmitter and a list of receivers, a snapshot is taken of the positions of all involved nodes. Then, for each transmitter-receiver combination a separate BELLHOP query is compiled using a two-dimensional slice marking the difference in depth between transmitter and receiver and their Euclidean horizontal distance. An example of such a plane is illustrated in Figure 4.4. Such an approach assumes no obstruction in the direct line-of-sight between the transmitter and receiver and it neglects refraction of sound out of the slice (horizontal gradients). However, as mentioned in section 2.2.3, these features are mostly



(a) Example setup with transmitter Black Box and receiver Relay node. The plane that serves as the slice between transmitter and receiver is marked yellow. This plane is forwarded to BELLHOP.

(b) How BELLHOP interprets the input from Figure 4.4a.

Figure 4.4: 3D Euclidean space conversion to 2D BELLHOP input

occurring in strong bathymetric features and it is an active area of research. After for every transmitter-receiver pair a BELLHOP query is compiled, UWALSim will execute BELLHOP for each query simultaneously in different threads, since a BELLHOP query is relatively slow compared to other much simpler propagation models. When all queries are finished, the threads are joined and the merged results are returned to the caller. Note that the BELLHOP propagation model is the most processing-power-demanding part of the simulator. The more nodes in the network, the larger actual time it takes to run the simulator for a specific amount of simulation time.

4.4.2 SimpleUW

Simple Underwater propagation model can calculate arrival times of signals without taking into account multipath effects and curved sound rays due to a difference in propagation speed on various depth levels. For every transmitter-receiver pair, SimpleUW calculates an averaged sound speed based on the positions of the transmitter and receiver. For determining the sound speed at the position of a node, it uses the Munk sound speed formula in equation 2.1. The averaged sound speed is used to estimate a time-of-arrival of a signal given the Euclidean distance between the transmitter and receiver. Even though this method is less accurate than the BELLHOP arrival time predictions, it is much faster to calculate. SimpleUW is only used for calculating arrival times of signals between acoustic modems for underwater communication purposes. Therefore, it only impacts the significance of the communication delay estimation, but not the localization accuracy.

4.4.3 SimpleTerrestrial

SimpleTerrestrial propagation model is similar to SimpleUW. Their only difference is that SimpleTerrestrial is meant for terrestrial radio-frequency communication and therefore uses a constant propagation speed of $3 \cdot 10^8$ m/s.

4.5 Communication

This section covers explanations of the acoustic modem implementation. Several topics are discussed, including signal collision avoidance and detection, modulation, FEC, noise on the channel and used medium access control schemes.

4.5.1 Acoustic channel

A channel in UWALSim represents a communication channel in a distinct frequency passband where transceivers can send messages over. Several types of channels exist, but here only the acoustic underwater channel is discussed since that is most relevant to this study. Subscribed transceivers will receive any message sent in the channel if they are in range, regardless whether they are recipient or not. It is the responsibility of the modem to discard messages that are meant for other transceivers. Signal interference is only modelled for transceivers belonging to the same channel, since each channel is non-overlapping. Transceivers will not be able to receive (or interfere with) messages from a channel other than their own. When a packet is sent over the acoustic channel, the channel calculates what other transceivers in the channel are in range for receiving this packet. Based on the SNR, the channel adds Additive White Gaussian Noise (AGWN) to the *raw_symbols* of the transmission (see section 4.5.2). Every symbol X_i in a stream of symbols is subject to independent noise Z_i, resulting in output Y_i, as shown in the following equation:

$$\begin{gathered} Y_i = X_i + Z_i \\ \text{with } Z_i \sim \mathcal{N}(0, 10^{\frac{-\text{SNR}}{20}}) \end{gathered} \tag{4.3}$$

where $\mathcal{N}(\mu, \sigma)$ is the Normal distribution with mean μ and standard deviation σ. The SNR (dB) is provided in UWALSim's configuration file.
The acoustic channel makes use of the SimpleUW propagation model to estimate the propagation time of the signal from transmitter to each receiver in the channel within range. Then, SimPy schedules a signal reception event to each receiver by utilizing the estimated propagation time.

This implementation uses a constant SNR for all acoustic modulated transmissions. The distance between transmitter and receiver is a significant parameter for the SNR. However, a constant SNR is justifiable due to a fixed distance between nodes in the relay link.

4.5.2 Acoustic transceiver

UWALSim includes an implementation of a generic acoustic transceiver, based on the characteristics and energy profiles of the EvoLogics R34D (Appendix C). Its default behavior is to listen for incoming signals except when it is transmitting. Its most important properties are discussed in this subsection.

Data packeting, FEC and modulation

When a data collection strategy requests the node to transmit a measurement to the node one level higher in the relay link, the measurement is converted to a data packet

according to the scheme illustrated in Figure 3.4. Since UWALSim is written in Python, built-in data types contain extra information and thus have increased instance sizes compared to data type sizes in C (what is assumed for the packeting scheme in this study). Therefore, directly obtaining the size of the packet structure in Python gives a unreliable value. The size of the packet influences the transmission time of the packet. To reliably simulate packet transmission times, UWALSim creates a random bytearray sequence matching the size of what a constructed packet would be without ECC bytes, and adds ECC bytes matching the content of the bytearray. The number of ECC bytes to add can be configured in UWALSim's configuration file. Reed-Solomon FEC is implemented utilizing the *reedsolo* Python package [38]. This package provides interfaces for encoding, decoding and verifying data on a Reed-Solomon codec utilizing a configurable amount of ECCs. Through the configuration file of UWALSim one can set a symbol size for the PSK constellation scheme. The random sequence of bytes is then encoded in a PSK scheme matching the configured symbol size, leading in a sequence of encoded random data called the *raw_symbols* of a measurement. The raw_symbols are actually subject to AWGN and it is used to detect demodulation errors after receiving a packet. When the packet is constructed, it is added to a buffer that stores all packets up for transmission. The buffer is processed when the acoustic transceiver enters its transmission slot.
For every packet that is up for transmission, the transmission time is calculated using the size of the packet and the available throughput. The available throughput is calculated using Nyquist's formula (equation 2.13) on a channel bandwidth of 5 kHz (section 3.3.1). When using a symbol size of 2, the available throughput is 10 kbps. After switching states, the transceiver will remain transmitting on the channel for as long as the transmission time of the packet. This process repeats for each packet up for transmission in the buffer, or until the transmission slot ended. It is the responsibility of the acoustic channel to schedule reception of each packet.

When a message arrives at a transceiver, it is first checked if the transceiver is listening to incoming signals (by checking if its state is 'listening'). If so, that means the transceiver is ready to receive the message. Then it checks if the transceiver is a recipient of the message. If so, it changes its state to 'receiving' and starts the full decoding process. It waits for the transmission time of the packet so the entire packet is in the receiver buffer. The receiver loads all pre-configured modulation configuration to its decoder and starts decoding the arrived raw_symbols of the packet, resulting in a bytearray of data that might have changed due to the AWGN in the channel. To verify the integrity of the arrived data, it is checked by the reedsolo codec if it can be decoded. This decoding process validates the data using the added ECC bytes. Next to serving as a checksum, if equation 2.12 is satisfied the codec is able to repair corrupted bytes using the ECC bytes. If decoding fails, the arrived message will not be forwarded to the parent node that equips the transceiver. If the message is successfully received, it is checked whether the message is a measurement or an acknowledgment. If it is a message, the transceiver generates an acknowledgment packet and puts it in the transmission queue. If the received packet is an acknowledgment, the transceiver updates its expected acknowledgments list accordingly.

Signal collision avoidance and detection

As discussed in section 2.4.6, there are several types of signal collisions to consider. This paragraph discusses how UWALSim detects or systematically prevents each type of collision. Note that the medium access control scheme as discussed in section 4.5.3 is responsible for avoiding these collisions.

- **TX-TX conflict**
  Since UWALSim processes scheduled transmissions on a transceiver one-by-one from the queue, it is not possible that multiple transmissions are scheduled to send simultaneously.

- **TX-RX conflict**
  A transceiver has several states (off, idle, listening, receiving and transmitting). It can only have one state at a time. When the node starts transmitting, it changes to the transmitting state. When a packet arrives at a particular moment, it is checked whether the receiver is in listening mode (meaning it listens for signals but it does not yet decode them). If not, the receiver is either receiving another message or it is transmitting. In both cases, the incoming message cannot be received and thus will not be processed.

- **RX-RX conflict and RX-interference**
  SimPy supports modelling preemptable operations that can only be invoked by one process at a time, similar to using semaphores. Receiving a message is modelled as such an operation. This ensures that only one message can be received at a time. If message b arrives while message a is being received, the simulator preempts the process of receiving message a and hence cancels further processing of both message b and a.

4.5.3 Medium access control scheme

UWALSim is equipped with a Medium Access Control module, where one or more MAC schemes can be implemented for creating and enforcing transmission schedules. The transmission slot scheduling as discussed in section 3.3.1 is implemented, where depending on the level in the relay link a fixed amount of transmission time is allocated for a node. Each node can independently calculate at what time its next transmission slot is scheduled and how long the slot is. Each relay link has an independent MAC module instance running in UWALSim. At the start of the simulation, the MAC module will start its transmission schedule with the relay node deepest in the relay link. At the start of a transmission slot for node n, the simulator informs the transceiver of n that its transmission slot has started, so the transceiver can start processing its transmission queue. This process of processing the transmission queue is modelled as a preemptable process, thus the simulator can interrupt the process at any time. When a transmission slot elapses for node n, the simulator will check if the process has finished on itself (e.g. when the transmission queue was fully processed). If not, the simulator interrupts the process, with the consequence that the packet being transmitted upon interrupting the process must be retransmitted in the next transmission slot. After waiting for the appropriate guard time, the simulator repeats this process for the next node in the transmission schedule.

4.6 Energy

One goal of UWALSim is the ability to measure energy consumption in the network. This is achieved by the Energy module, which consists of two major parts: *Energy source* and *Energy consumer*. Each of the parts is discussed in this section.

4.6.1 Energy source

An energy source is a container holding a finite amount of resources. In the simulator, the container is represented by a battery and the resource is the battery capacity. When constructing a battery, one must provide the battery capacity (mAH), the battery voltage (V) and the initial capacity of the battery (defaults to fully charged). Every node in the network is equipped with a battery, whose properties can be configured by means of a special energy properties file as illustrated in Appendix E.2. Energy consumers can consume resources (mW) from this battery for a certain amount of time (s). In the simulator, one can configure that a battery can die. This means that the node holding the battery will be informed about the battery state when the battery has no more resources left. The node will then turn off all devices that require power and therefore it will quit its operations. However, for monitoring purposes, it can be configured that a battery keeps on providing energy even after it ran out of resources. In this case, when a battery has no more resources left, it will not inform the node about its changed state. This model of a battery assumes the battery capacity decreases linearly over the amount of consumed resources. Furthermore, it does not support battery degradation over time or fluctuating effective capacity based on temperature.

4.6.2 Energy consumer

An energy consumer is responsible for consuming energy from an energy source for particular device operations or device states. Every device that consumes power has a power consumption profile attached to each state the device can have, as can be seen in Appendix E.2. When a device switches its state (e.g. an acoustic modem changes state from *listening* to *transmitting*), the energy consumer starts consuming the amount of energy matching the new device state.

Once the energy consumer starts, it consumes energy from the energy source at two types of moments: scheduled and upon stopping. Scheduled energy consumption takes place every second. The energy that the devices consumes in a second is subtracted from the energy source. After subtracting energy, the energy consumer records the time of the last consumption and schedules the next consumption to occur in 1 second. When an energy consumer is stopped (when a device switches state) it calculates the amount of energy the device has consumed in the time between the last consumption and when it stopped consuming. This energy is subtracted from the energy source and the next scheduled energy consumption is cancelled. The energy consumer assumes a constant energy consumption during a device state.

4.7 Monitoring

Monitors are classes that utilize *Pandas DataFrames* to keep record of various types of data throughout the simulation. There exist several types of monitors:

- **Battery monitor**
  For each node, the remaining battery capacity is monitored over time. Also, each battery transaction is logged.

- **Ping monitor**
  All ULB transmissions are tracked here from transmission until localization of the source. Reasons for dropping ping measurements are listed here as well, indicating how many measurements actually arrived on the head anchor. This is the most important monitor as it lists the localization accuracy and the end-to-end delay for each ULB transmission.

- **Self-positioning monitor**
  For every node in the network, this monitor captures all self-positioning queries and lists the error on self-positioning over time.

- **Acoustic packet monitor**
  Every acoustic packet is tracked here. The status of the transmission, the communication delay and the cause of packet loss (if applicable) are registered as well.

- **RF packet monitor**
  Every radio-frequency packet is tracked here. The communication delay is registered as well.

Each type of monitor is constructed *once* by SimCore upon initializing the simulation. What types of monitors to enable can be configured in the simulator. SimCore is globally accessible by all UWALSim modules, therefore it is easy to retrieve a monitor instance to record information in every step of the simulation process.

A full overview of the attributes that each monitor type records can be found in Appendix D.

Chapter 5

Experiments

5.1 Scenarios

The approach discussed in section 3 is evaluated using UWALSim simulations (chapter 4). To increase the confidence in the experimental performance of this approach, several scenarios are simulated. Each scenario is discussed briefly in this section, whereas specific changes in configuration for each scenario are listed in Appendix F.2. Involved energy consumption profiles can be found in Appendix E.2. In every scenario, the network is deployed in an oceanic environment that has no coastal area in the near surroundings and the ocean depth is fixed to 6 km. Once the blackbox reaches the ocean bottom, the simulation ends since the wreckage does no longer move when it has reached the ocean floor. Ocean currents depend on wind force, tides, the earth's rotation, the sun and water density. Hence the direction and magnitude of such currents depend on the crash location and time. Therefore, for all simulation scenarios an averaged current profile is used which can be found in Appendix F.1. Depending on the aircraft model and crash impact, the wreckage (and therefore the ULB) may float for an amount of time. Several simulation scenarios consider a floating time of 20 minutes before the wreckage starts sinking. There are also scenarios where the wreckage is assumed to sink immediately after having crashed. All network configurations are conform to section 3.1 where there exist 3 anchor nodes, and each relay link consists of 4 relay nodes. Since all nodes that are part of the network origin from the same crashed aircraft, they start close together. By means of an onboard motor, anchor nodes can displace themselves to a location relative to the head anchor, together forming a triangle. When an anchor node reaches its desired destination, it stops the onboard motor and floats with the ocean current. Immediately at the start of a simulation anchor nodes start dropping relay nodes. When the distance between the latest dropped relay node and its anchor node exceeds 1250 meters, the anchor nodes start deploying their next relay node. When a relay node is dispatched, it synchronizes with its anchor on time and self-positioning.

Simulations where the ULB has a ping interval of 1 second and the network calculates a location for every received ping have a runtime of 1 to 2 full days (depending on the processing power of the host running UWALSim). These long runtimes are due to the computational complexity introduced by the BELLHOP propagation model and the necessity of using an offset size of 3 to satisfy equation 3.2. Due to time constraints and the preference of simulating a wide range of different scenarios, it is decided to simulate for a ULB ping interval of 10 seconds. It is expected that scenarios with a ULB ping interval of 1 second have a similar localization performance.
Each scenario has three variations concerning self-positioning accuracy. The first being 'moderate' self-positioning, the second being 'perfect' self-positioning and the third being

'bad' self-positioning. The configured error values belonging to each variation are listed in Appendix F.2. Since it is expected that self-positioning has a significant impact on localization error, every scenario is run for each of these variations. Table 5.1 describes the different scenarios:

Scenario	**BlackBox location relative to triangle of nodes**	**Float time of BlackBox**	**Sinking speed of BlackBox relative to Relay node deployment**	**Sound speed profile underwater used in BELLHOP**
1	Outside triangle	20 minutes	BlackBox sinking: 1 m/s Relay nodes sinking: 1.5 m/s	Munk profile
2	Outside triangle	Immediately sinking	BlackBox sinking: 1 m/s Relay nodes sinking: 1.5 m/s	Munk profile
3	Inside triangle	20 minutes	BlackBox sinking: 1 m/s Relay nodes sinking: 1.5 m/s	Munk profile
4	Inside triangle	Immediately sinking	BlackBox sinking: 1 m/s Relay nodes sinking: 1.5 m/s	Munk profile
5	Outside triangle	20 minutes	BlackBox sinking: 2 m/s Relay nodes sinking: 1.5 m/s	Munk profile
6	Outside triangle	Immediately sinking	BlackBox sinking: 2 m/s Relay nodes sinking: 1.5 m/s	Munk profile
7	Outside triangle	20 minutes	BlackBox sinking: 1 m/s Relay nodes sinking: 1.5 m/s	Constant speed of 1496 m/s
8	Outside triangle	Immediately sinking	BlackBox sinking: 1 m/s Relay nodes sinking: 1.5 m/s	Constant speed of 1496 m/s

Table 5.1: Descriptions of scenarios used for running experiments.

5.2 Localization

Figure 5.1 shows how each variation on self-positioning configurations relate to an error on self-positioning at the end of the simulation for each relay node. Perfect self-positioning always has an error in self-positioning of 0 m. As can be seen from the figure, the first-deployed relay nodes from each relay link suffer the greatest self-positioning error. A moderate self-position performance has errors up to 25 m, whereas a bad self-position performance has errors up to a couple of 100 meters. Most part of this large error originates from the first 15 minutes of simulation, where the anchor nodes travel from their deployed positions to their desired positions such that the network forms a triangle of anchor nodes. Other self-positioning errors are introduced due to horizontal displacement by ocean currents. *Relay 3-1* has the largest self-positioning error compared to other relay nodes that were deployed first. It has a larger error due to its anchor node travelling the fastest to its desired position since it was travelling along with the ocean current. *Relay 2-2* has on average a larger self-positioning error than other relay nodes that were deployed second in their relay link. This is the case because the anchor node of relay link 2 was still travelling to its desired location when it deployed its second relay node in the ocean. Figure 5.2 illustrates the localization performance of this approach in the scenarios presented in the previous section. Following research question 1 from section 1.2, an error-margin of 50 meters is used to mark a location estimate as being successful. Appendix G.1 illustrates the localization error per scenario and variation over time.

As can be seen from Figure 5.2, the localization error is strongly related to the quality of the self-positioning of the nodes. For the 'bad self-positioning' variant, no scenario shows

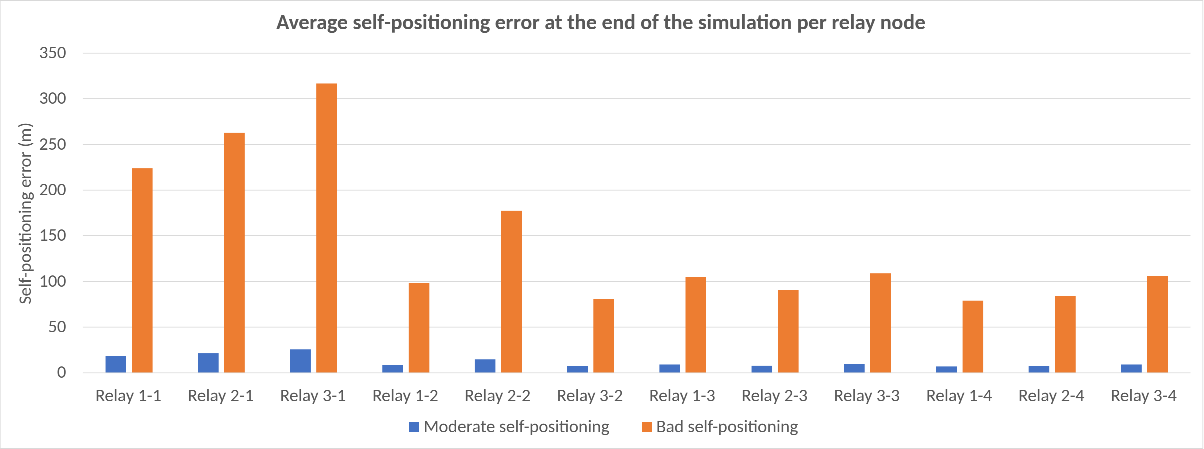


Figure 5.1: Average self-positioning error at the end of each simulation per relay node *x*-*y* where *x* represents the relay link number and *y* represents the order of deployment in the relay link (for instance, *Relay 1-1* is the relay node in relay link 1 that was deployed first).

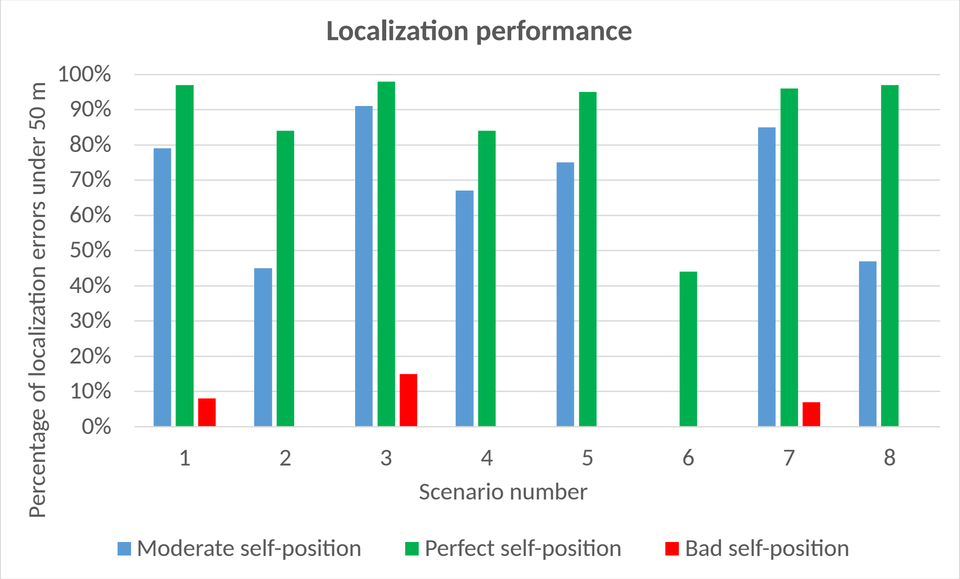


Figure 5.2: Localization performance of each scenario and self-positioning variation.

an acceptable performance of this approach. For most scenarios with the moderate self-positioning variation, the localization error is often largest at the beginning and at the end of the simulation run. The error at the beginning is due to a false time-of-arrival estimation of ping signals arriving on anchor node hydrophones. Since anchor nodes float, in situations where the black box is on near-surface depth it is very unlikely that a direct line-of-sight sound ray is captured by hydrophones of the anchor node. When there exists a dynamic speed of sound, sound rays tend to curve as illustrated in Figure 2.2. If anchor nodes receive pings from a black box at near-surface depth, it is most likely subject to a surface bounce. Since there are no direct line-of-sight inbound signals that have a larger amplitude, the data collection strategy thinks that the surface-bounced signal would be a direct line-of-sight sound ray. Once the black box reaches a depth of approximately 200 meters, in most scenarios the localization of the black box stabilizes. The large error at the end of the simulation occurs when the black box reaches depths of over 5 km. Then, in most scenarios only 6 hydrophones receive black box pings. The other hydrophones are out of range for receiving the black box signals. The remaining 6 hydrophones that do receive these signals, are also the relay nodes that have the largest self-positioning error. Hence, the combination of few signal receptions and bad self-positioning lead to larger errors in localization of the black box at large depths.
All graphs in Appendix G.1 show throughout the simulation run clear outliers in localization error. Its origin can have two reasons. The first reason could be that new relay nodes recently have been deployed in the ocean. Their position is very similar to the position of their anchor nodes. The Moore-Penrose pseudoinverse least-squares method tends to give bad solutions when there are nodes sharing approximately the same location and time-of-arrival value for black box pings. Small measurement errors then lead to a large difference in estimated position of the source. The second reason could be that at the point of the error peak one or more nodes moved out of range for receiving black box pings. When a node no longer receives new pings from the black box, it is possible that it still registers some bounced multipath arrivals from previous ULB transmissions when it still was in range. As long as a node receives these 'echoes' from previous transmissions, for each time window it will forward the measurement with the largest amplitude to the head anchor for localization as if it were direct line-of-sight measurements from a recent ULB transmission. For the head anchor, this outlier is hard to detect and causes large deviations in source localization. These location estimations do have a large residual sum of squares, meaning that the solution is not a good fit to the input and therefore it is possible for the head anchor to filter out these false solutions based on the value of the residual sum of squares.

It can also be observed from Figure 5.2 and Appendix G.1 that scenarios where the black box sinks immediately have a worse localization performance than scenarios that have an equivalent setting except that the black box first floats for 20 minutes. It takes longer to stabilize the black box localization. In situations where the black box starts sinking after 20 minutes, the network has already deployed 2 relay nodes per relay link. At this point in time, all hydrophones are able to capture black box signals, hence there are more measurements to calculate a source location with. Also, there is a larger diversity in hydrophone depth locations, which contributes to better location estimations. When the black box starts sinking immediately, there is only 1 deployed relay node per relay link. This means that there are less measurements per black box ping and the deployed nodes all share similar depth levels. The performance on the black box localization starts to improve after the second set of relay nodes have been deployed and this set has reached a sufficient depth difference relative to their anchor nodes.

When comparing the scenario where a black box is positioned outside of the anchor-triangle (scenario 1 and 2) to the scenario where the black box is positioned inside the anchor-triangle (scenario 3 and 4), it does not show a significant difference in localization performance. Scenario 3 and 4 have a slight performance improvement over scenario 1 and 2 respectively because in scenario 3 and 4 on average more hydrophones capture black box signals. From all transmitted black box pings, the average number of hydrophones that captured these pings is 10.6 hydrophones for scenario 1 and 11.0 hydrophones for scenario 3.

From scenario 5 it can be observed that localization performance is not significantly affected when the black box sinks slightly faster than the deployed relay nodes, provided that the black box floats for 20 minutes before it starts sinking. However, from scenario 6 it becomes evident that partial network deployment is crucial when the black box sinks faster than the relay nodes. If both start sinking at the same time, the distance between the deepest relay node and the black box will increase until the black box has reached the ocean bottom. On average, each black box ping is only received by 6.6 hydrophones. Similar to the increased localization errors near the end of each simulation run as discussed earlier in this section, these errors are large due to a limited number of hydrophones that detect the black box signals and the nodes that do register black box pings are also the nodes with the worst self-positioning.

When comparing scenario 7 and 8 to scenario 1 and 2, one can observe that the localization performance is not significantly affected by the assumption that the sound speed underwater is constant. The percentage of location estimations under 50 m error is for both situations very similar, but for scenario 7 the average localization error among the measurements that have an error under 50 m is 11.8 m (moderate self-positioning), whereas for scenario 1 this average lies on 21.8 m. For the perfect self-positioning variants of scenario 7 and 8, this average approximates 0 m error because self-positioning and sound speed errors are the most significant factors for errors in source localization.

5.3 Communication

For all simulations, a BPSK modulation scheme as introduced in section 2.4.2 is used for data transfer over the acoustic channel. Simulation results support the claim made in section 3.3.1 that a BPSK scheme (symbol size of 2) provides sufficient data throughput on each channel with a 5 kHz bandwidth. Because of the utilized symbol size of 2, the modulation scheme is more robust against noise on the channel. In every scenario and variation, all packets showed a symbol-to-error rate of 0, meaning that all symbols were correctly demodulated. Because each acoustic transmission was completely error-free, the error correction codes (ECCs) encoded in the header of each packet only served as a checksum. There has been no case where forward error correction (FEC) prevented a packet from being retransmitted as there were no corrupted arrivals. The threshold where noise starts to impact the symbol-to-error rate on the utilized modulation scheme lies on a SNR of 8 dB. Here, by using the ECCs the wrongly demodulated bits can

be corrected. From a SNR of 4 dB or lower, the number of wrongly demodulated bits per packet due to noise on the channel becomes too large for the FEC to be able to repair the message. This would lead to frequent retransmissions which increases battery consumption, communication delay and when packets are dropped due to exceeding the maximum amount of retransmissions this affects localization performance too.
For all simulated scenarios and variations, there have been no cases of signal collisions of any kind. This supports the proposed transmission schedule as introduced in section 3.3.1 to be a valid transmission schedule for this application. Every time a new relay node was deployed in the ocean, the transmission schedule was extended with another node on the channel. Because the deployment of new relay nodes in the ocean happens at fixed moments in time, other nodes can independently (since they are time-synchronized) alter their transmission schedule according to the number of relay nodes on the channel.

5.4 Energy consumption

Each node is equipped with a fully charged battery. Since this setup does not allow battery recharge, the lifetime of the network is finite. Research question 3 (section 1.2) states that the network must operate for at least 4 hours, as the black box is expected to definitely have reached the ocean bottom after this time. As can be seen in Figure 5.4, in scenarios where the ULB has a ping interval of 1 s and the network is configured to localize every received ping, the network lifetime is too short as the first node disconnects after 48 minutes. Therefore, there is need to adapt the behavior of the network to extend battery life. Figure 5.3 shows how each type of node spends its energy resources. As can be seen from the figure, relay nodes spend by far most energy on acoustic data transmissions. Therefore, most impact on extending battery life can be achieving by reducing acoustic transmissions. Furthermore, one can see from Figure 5.4 that *Anchor 1* has the least expected battery lifetime from all anchor nodes. This is the case because in the simulated scenarios *Anchor 1* was travelling for the longest time towards its desired position to form the anchor-triangle. Given that using the onboard motor takes most of the energy resources, this affects battery life significantly. Note that anchor nodes only travel to their desired location once, so they can last longer on the long run assuming they don't have to use their motors again. From the figure it also becomes evident that the first-deployed relay nodes have the longest expected battery life. Even though they operate longer than relay nodes that were deployed later, these nodes do not have to forward messages from nodes below them in the relay link. Since acoustic transmissions are the most expensive operations in terms of battery life, not forwarding messages saves much energy.
To increase the network's operating life to at least 4 hours, a change in network behavior must be made. The proposed change is that each node does not process 9 out of 10 received ULB pings. This is achieved by only forwarding signal peaks from the first out of ten time windows in the data collection strategy. The measurements that are forwarded can still be used to localize the black box since time windows are synchronized amongst all nodes. Figure 5.4 illustrates how the network lifetime is affected by this change in configuration. As can be seen from the figure, the network lifetime has been extended to over 6 hours which satisfies the requirement in research question 3. The downside of this approach is that a new location of the black box is obtained every 10 seconds, instead of every 1 second.

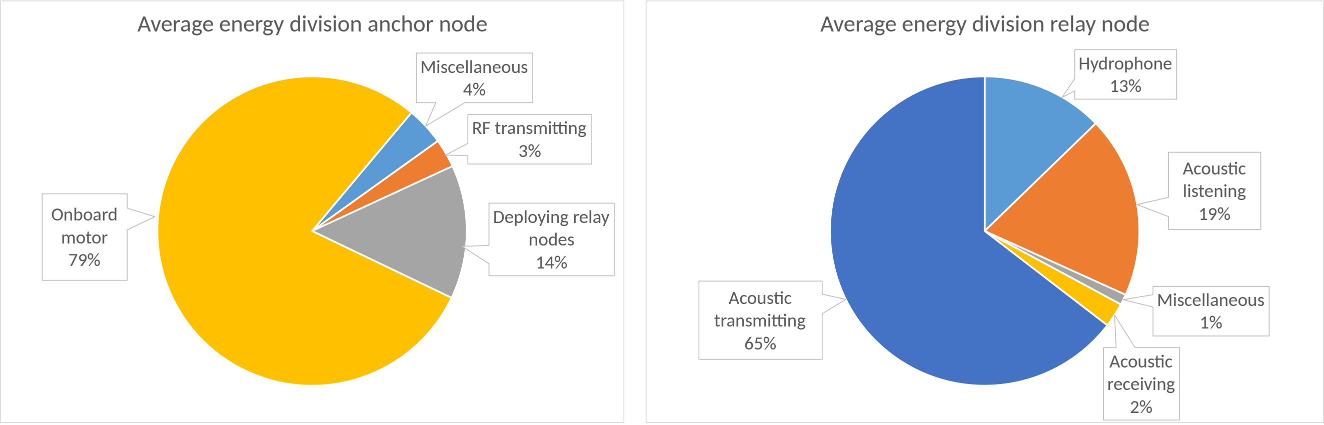


Figure 5.3: Average energy division in a simulation run for anchor nodes and relay nodes in scenarios with a ULB ping interval of 1 s and every received ping is processed for source location estimation.

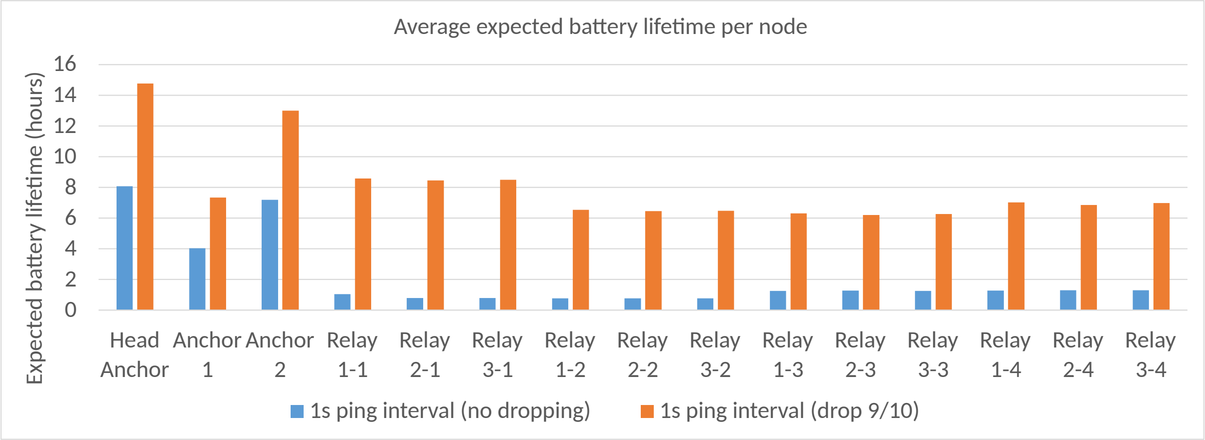


Figure 5.4: Average expected battery life per node for a ULB ping interval of 1 s with and without dropping measurements for saving battery life. Relay nodes are identified as Relay x-y where x represents the relay link number and y represents the order of deployment in the relay link.

Chapter 6

Discussion and conclusion

The possibility of successfully localizing a flight recorder after an oceanic flight crash by using an onboard ad-hoc wireless sensor network has been studied. When designing and evaluating the approach, the most important characteristics of the underwater acoustic medium have been taken into account such as a dynamic sound speed profile, low propagation speed, multipath effects and noise. A novel wireless sensor network structure consisting of anchor and relay nodes has been used to extend the reception of ULB signals up to 6 km depth. For node-to-node communication, a custom data packeting structure has been designed to minimize the required data throughput in the slow underwater acoustic communication medium with large propagation delays. To avoid signal collisions when transmitting data over the acoustic channel, a custom transmission scheduling approach has been tailored to the network's hierarchy and channel characteristics.

To verify that the proposed approach meets the requirements set in research questions 1-3, a custom simulator has been developed that utilizes dedicated propagation models for realistic underwater sound propagation modelling. Also, the simulator is able to detect signal collisions, simulate noise on the acoustic channel and it models energy consumption. The results of several simulated scenarios have been presented. From these results, it becomes clear that the performance of this approach strongly depends on the quality of receiver inertial self-positioning. When considering a 'moderate' self-positioning performance of errors up to 25 meters, the localization performance of this approach depends on how far the network has been deployed in the ocean when the flight recorder starts sinking. In cases where the network has been deployed to a depth of at least 1500 meters at the moment the flight recorder starts sinking, at least 75% of all processed ULB pings lead to localization of the flight recorder under 50 meters of error. Good to remark here is that the localization performance is based on scenarios where the ULB has a 10-second ping interval, because simulating on a 1-second ULB ping interval would take too much time to include in this study due to an increased computational complexity for source localization. It is however expected that localization performance would not change significantly when simulating on 1-second ping intervals. Even though localization can be off when the flight recorder starts sinking and when it exceeds depths of 5 km, the sinking trajectory of estimated positions with high confidence could be used to aid for correction of false flight recorder position estimates. The epoch problem that was introduced due to synchronously listening to ULB pings at different locations on a channel with a low propagation speed has been resolved, even though it introduces a large computational complexity when the ULB pings at an interval of 1 s. To limit computational complexity, one could re-use information from previous high-confidence solutions to eliminate permutations of measurement sets for new incoming measurements. Alternatively, one could perform the localization calculations on a remote high-power processor by forwarding measurements via the satellite link on the anchor nodes.

Since this approach does not require a large data throughput over the underwater acoustic channel, modulation can be optimized for robustness against noise instead of maximizing data throughput. Simulation results support that this approach is robust against noise on the acoustic communication channel with a SNR of 5 dB or better. It is however unclear if a larger data throughput by means of utilizing a higher-order modulation scheme would lead to significant energy savings for relay nodes since transmission times would decrease.
Simulation results indicate that this approach would not be able to operate for at least 4 hours in scenarios where the ULB has a ping interval of 1 s, provided that the network processes every received ping. To extend battery life, the approach could be adapted to only process one out of ten received ULB pings. Simulation results show that this adaptation to the original approach can extend the network life to over 6 hours. One could also argue to change the design of the ULB to ping on a 10-second interval. By applying this change, the ULB will be able to ping for a longer time, the computational complexity for solving the epoch problem will be reduced significantly and there is no need to drop ping measurements to save battery life.

To conclude, the approach proposed in this study outperforms the current state-of-the-art in localizing flight recorders after oceanic crash events in two ways. First, by installing this approach in an aircraft the first reliable flight recorder location estimations are obtained in 5 to 25 minutes (depending on the scenario) after the first ULB ping has been transmitted. Current state-of-the-art solutions are not to be embedded in an aircraft and require the search equipment to travel to an alleged crash site first. Second, the detection range of the flight recorder's ULB is increased significantly. By installing this approach, ULB pings originating from deep-ocean levels can be detected too as opposed to a single array of hydrophones used in many state-of-the-art solutions.

6.1 Future work and recommendations

Several scenarios were taken into account for assessing the performance of this approach. From experimental results, a number of recommendations should be taken into account when designing a network as described in this study. First of all, accurate inertial self-positioning systems for relay nodes are essential for optimal flight recorder localization performance. A suggestion for improving self-positioning estimation is to limit the horizontal displacement of a relay node relative to its anchor by for instance reinforcing the string that physically connects relay nodes. Second, experimental results show that flight recorder localization improves when the network is already partially deployed when the flight recorder starts sinking. It is therefore suggested to start deploying relay nodes immediately after anchor nodes hit the water. Relay nodes could be designed to sink quickly by further decreasing the buoyancy of the object. This way, the network could be deployed faster.

In future work, more scenarios of oceanic flight crashes can be taken into account for assessing the performance of this approach. Specifically scenarios with a ULB ping interval of 1 s should be considered for verifying the results in section 5.2 that currently assume a ULB ping interval of 10 s. Also, one could research what network setup and configuration optimizes localization performance. Nonetheless the computational complexity for calculating a source location is likely to be reducible by defining constraints to plausible permutations of measurement sets, by for instance taking into account the position of nodes relative to each other and information about node displacement. Furthermore it would be interesting to analyze how information from past location estimations of the flight recorder could improve new flight recorder location estimations. Finally, it would be interesting to research how well this approach integrates into other applications that require similar technology, for instance whale tracking, submarine detection or localization of divers.

Bibliography

[1] RP Acoustics. *Fiber Optic Probe Hydrophone FOPH 2000.* URL: https://rp-acoustics.com/english/Fiber-Optic-Probe-Hydrophone-FOPH-2000.htm.

[2] Elizabeth Ann Alvanas. “A Line Array of Directional Hydrophones for Improved Detection of Emergency Locator Beacons”. MA thesis. University of Rhode Island, elizabeth.alvanas@cox.net, 2018.

[3] Rafael Barmak et al. “Underwater Locator Beacon signal propagation on tropical waters”. In: (2017), pp. 1–4. DOI: 10.1109/RIOAcoustics.2017.8349738.

[4] Mandar Chitre. *ArlPy - Toolbox for Underwater Acoustics - GitHub page.* URL: https://github.com/org-arl/arlpy.

[5] Mandar Chitre et al. “Recent advances in underwater acoustic communications amp; networking”. In: *OCEANS 2008.* Vol. 2008-Supplement. 2008, pp. 1–10. DOI: 10.1109/OCEANS.2008.5289428.

[6] Bureau d’Enquêtes et d’Analyses. *Final Report on the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro - Paris.* 2009. URL: https://bea.aero/docspa/2009/f-cp090601.en/pdf/f-cp090601.en.pdf.

[7] Roee Diamant and Lutz Lampe. “Underwater Localization with Time-Synchronization and Propagation Speed Uncertainties”. In: *IEEE Transactions on Mobile Computing* 12.7 (2013), pp. 1257–1269. DOI: 10.1109/TMC.2012.100.

[8] Digi. *Digi XBee 900MHz RF modems data sheet.* URL: https://www.digi.com/resources/library/data-sheets/ds_xtend.

[9] J. Dunlop and D.G. Smith. *Telecommunications Engineering, 3rd Edition.* Taylor & Francis, 1994. ISBN: 9780748740444. URL: https://books.google.nl/books?id=-kyPyn3Dst8C.

[10] EvoLogics. *EvoLogics Underwater Acoustic Modems product information guide.* URL: https://evologics.de/web/content/15634.

[11] Guangjie Han et al. “Localization Algorithms of Underwater Wireless Sensor Networks: A Survey”. In: *Sensors* 12.2 (Feb. 2012), pp. 2026–2061. ISSN: 1424-8220. DOI: 10.3390/s120202026. URL: http://dx.doi.org/10.3390/s120202026.

[12] Anders Heerfordt, Bertel Møhl, and Magnus Wahlberg. “A wideband connection to sperm whales: A fiber-optic, deep-sea hydrophone array”. In: *Deep Sea Research Part I: Oceanographic Research Papers* 54.3 (2007), pp. 428–436. ISSN: 0967-0637. DOI: https://doi.org/10.1016/j.dsr.2006.12.003. URL: https://www.sciencedirect.com/science/article/pii/S0967063706003268.

[13] A. Hurrell and P. Beard. “19 - Piezoelectric and fibre-optic hydrophones”. In: *Ultrasonic Transducers.* Ed. by K. Nakamura. Woodhead Publishing Series in Electronic and Optical Materials. Woodhead Publishing, 2012, pp. 619–676. ISBN: 978-1-84569-989-5. DOI: https://doi.org/10.1533/9780857096302.3.619. URL: https://www.sciencedirect.com/science/article/pii/B978184569989550019X.

[14] Mohammed Jouhari et al. "Underwater Wireless Sensor Networks: A Survey on Enabling Technologies, Localization Protocols, and Internet of Underwater Things". In: *IEEE Access* 7 (2019), pp. 96879–96899. DOI: 10.1109/ACCESS.2019.2928876.

[15] Wouter A.P. van Kleunen. "Echoes from the deep: Communication Scheduling, Localization and Time-Synchronization in Underwater Acoustic Sensor Networks". PhD thesis. Centre for Telematics and Information Technology, University of Twente, 2014.

[16] Vitaliy Kunin et al. "3D direction of arrival estimation and localization using ultrasonic sensors in an anechoic chamber". In: *2011 IEEE International Ultrasonics Symposium*. 2011, pp. 756–759. DOI: 10.1109/ULTSYM.2011.0184.

[17] Shuangshuang Li, Haixin Sun, and Hamada Esmaiel. "Underwater TDOA Acoustical Location Based on Majorization-Minimization Optimization". In: *Sensors* 20.16 (2020). ISSN: 1424-8220. DOI: 10.3390/s20164457. URL: https://www.mdpi.com/1424-8220/20/16/4457.

[18] Steven Li. *TDOA Acoustic Localization*. 2011. URL: https://s3-us-west-1.amazonaws.com/stevenjl-bucket/tdoa_localization.pdf.

[19] Daniel E. Lucani, Milica Stojanovic, and Muriel Medard. "On the Relationship between Transmission Power and Capacity of an Underwater Acoustic Communication Channel". In: *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*. 2008, pp. 1–6. DOI: 10.1109/OCEANSKOBE.2008.4531073.

[20] Wang J. Meng Z. Chen W. "Recent Progress in Fiber-Optic Hydrophones". In: *Photonic Sensors* 11.1 (2021), pp. 109–122. DOI: 10.1007/s13320-021-0618-5.

[21] Don Harris Mingwei Wang Richard Lane and Chen Li. *Ejectable flight data recorder systems, methods, and devices*. Patent ID US9738398B1. 2017.

[22] Walter H. Munk. "Sound channel in an exponentially stratified ocean, with application to SOFAR". In: *The Journal of the Acoustical Society of America* 55.2 (1974), pp. 220–226. DOI: 10.1121/1.1914492. eprint: https://doi.org/10.1121/1.1914492. URL: https://doi.org/10.1121/1.1914492.

[23] Mikhail Y. Plotnikov et al. "Thin Cable Fiber-Optic Hydrophone Array for Passive Acoustic Surveillance Applications". In: *IEEE Sensors Journal* 19.9 (2019), pp. 3376–3382. DOI: 10.1109/JSEN.2019.2894323.

[24] Michael B. Porter. *The BELLHOP Manual and User's Guide*. 2011. URL: http://oalib.hlsresearch.com/Rays/HLS-2010-1.pdf.

[25] Michael B. Porter. *The BELLHOP3D User Guide*. 2016. URL: https://usermanual.wiki/Document/Bellhop3D20User20Guide202016725.1524880335/html.

[26] I. S. Reed and G. Solomon. "Polynomial Codes Over Certain Finite Fields". In: *Journal of the Society for Industrial and Applied Mathematics* 8.2 (1960), pp. 300–304. DOI: 10.1137/0108018. eprint: https://doi.org/10.1137/0108018. URL: https://doi.org/10.1137/0108018.

[27] Thomas D. Rossing. *Springer Handbook of Acoustics*. 2014. DOI: 10.1007/978-1-4939-0755-7. eprint: https://doi.org/10.1007/978-1-4939-0755-7. URL: https://doi.org/10.1007/978-1-4939-0755-7.

[28] D. Brookes S. Wong R. Jassemi-Zargani and B. Kim. *Passive target localization using a geometric approach to the time-difference-of-arrival method.* 2017. URL: https://publications.gc.ca/collections/collection_2019/rddc-drdc/D68-2-079-2017-eng.pdf.

[29] Stefan Scherfke. *SimPy Discrete Event Simulation Engine - GitLab page.* URL: https://gitlab.com/team-simpy/simpy/blob/master/docs/index.rst.

[30] Henrik Schmidt and W. A. Kuperman. "Spectral and modal representations of the Doppler-shifted field in ocean waveguides". In: *The Journal of the Acoustical Society of America* 96.1 (1994), pp. 386–395. DOI: 10.1121/1.410489. eprint: https://doi.org/10.1121/1.410489. URL: https://doi.org/10.1121/1.410489.

[31] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[32] B.S. Sharif et al. "A computationally efficient Doppler compensation system for underwater acoustic communications". In: *IEEE Journal of Oceanic Engineering* 25.1 (2000), pp. 52–61. DOI: 10.1109/48.820736.

[33] Emmanuel Skarsoulis et al. "Underwater Acoustic Pulsed Source Localization with a Pair of Hydrophones". In: *Remote Sensing* 10 (June 2018), p. 883. DOI: 10.3390/rs10060883.

[34] Mariusz Specht. "Consistency analysis of global positioning system position errors with typical statistical distributions". In: *Journal of Navigation* 74.6 (2021), pp. 1201–1218. DOI: 10.1017/S0373463321000485.

[35] Sibo Sun et al. "Underwater Acoustical Localization of the Black Box Utilizing Single Autonomous Underwater Vehicle Based on the Second-Order Time Difference of Arrival". In: *IEEE Journal of Oceanic Engineering* 45.4 (2020), pp. 1268–1279. DOI: 10.1109/JOE.2019.2950954.

[36] A. A. Syed, W. Ye, and J. Heidemann. "T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks". In: *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications.* 2008, pp. 231–235. DOI: 10.1109/INFOCOM.2008.55.

[37] William H. Thorp. "Analytic Description of the Low-Frequency Attenuation Coefficient". In: *The Journal of the Acoustical Society of America* 42.1 (1967), pp. 270–270. DOI: 10.1121/1.1910566. eprint: https://doi.org/10.1121/1.1910566. URL: https://doi.org/10.1121/1.1910566.

[38] Stephen Karl Larroque Tomer Filiba. *A pure-python universal errors-and-erasures Reed-Solomon Codec.* URL: https://github.com/tomerfiliba/reedsolomon.

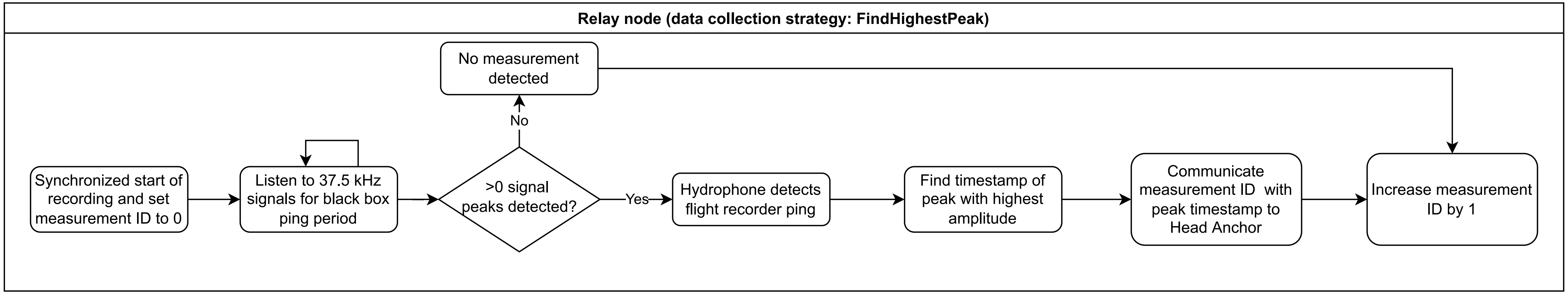
[39] Australian Transport and Safety Bureau. "Black box flight recorders - Fact sheet". In: ed. by Australian Government. URL: https://www.atsb.gov.au/media/4793913/Black%5C%20Box%5C%20Flight%5C%20Recorders%5C%20Fact%5C%20Sheet.pdf.

[40] Robert J Urick. *Principles of underwater sound.* 1983. ISBN: 0070660875 9780070660878.

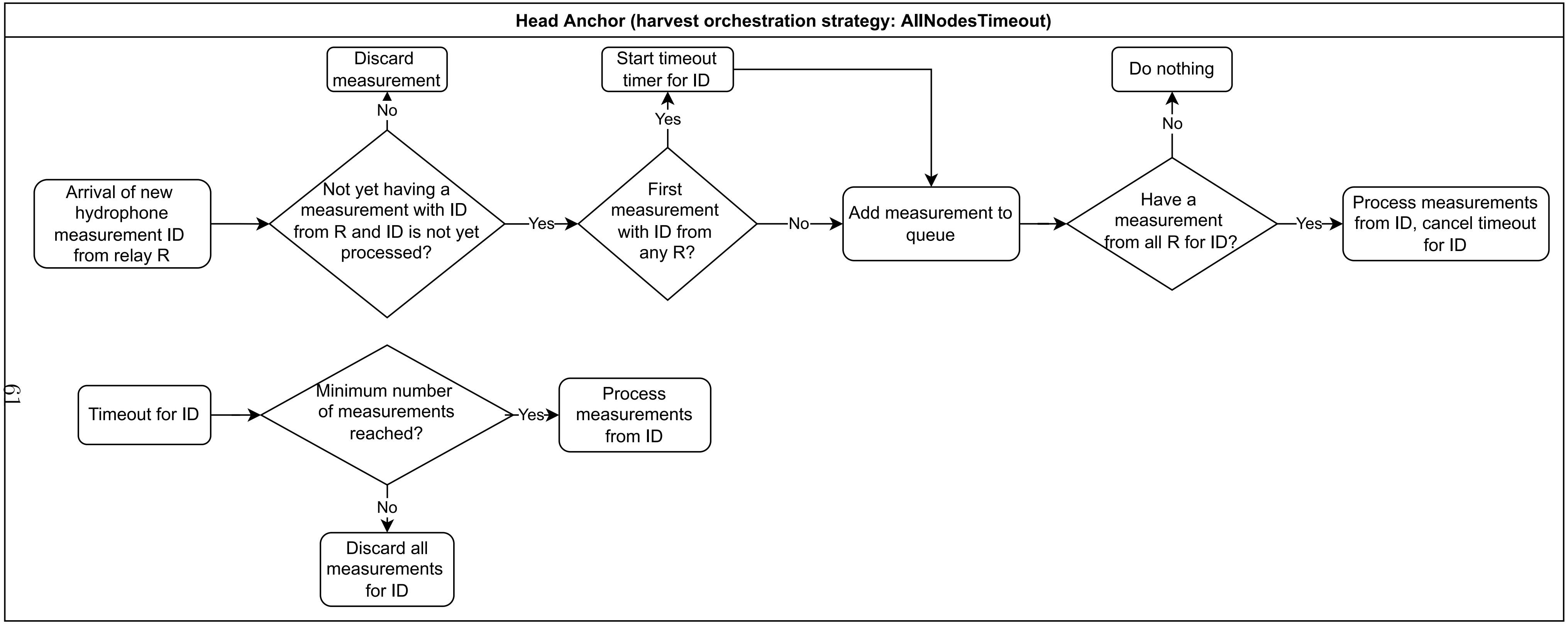
Appendix A

Localization process flowcharts

A.1 FindHighestPeak data collection strategy


Relay node (data collection strategy: FindHighestPeak)
No measurement detected
No
Synchronized start of recording and set measurement ID to 0
Listen to 37.5 kHz signals for black box ping period
>0 signal peaks detected?
Yes
Hydrophone detects flight recorder ping
Find timestamp of peak with highest amplitude
Communicate measurement ID with peak timestamp to Head Anchor
Increase measurement ID by 1

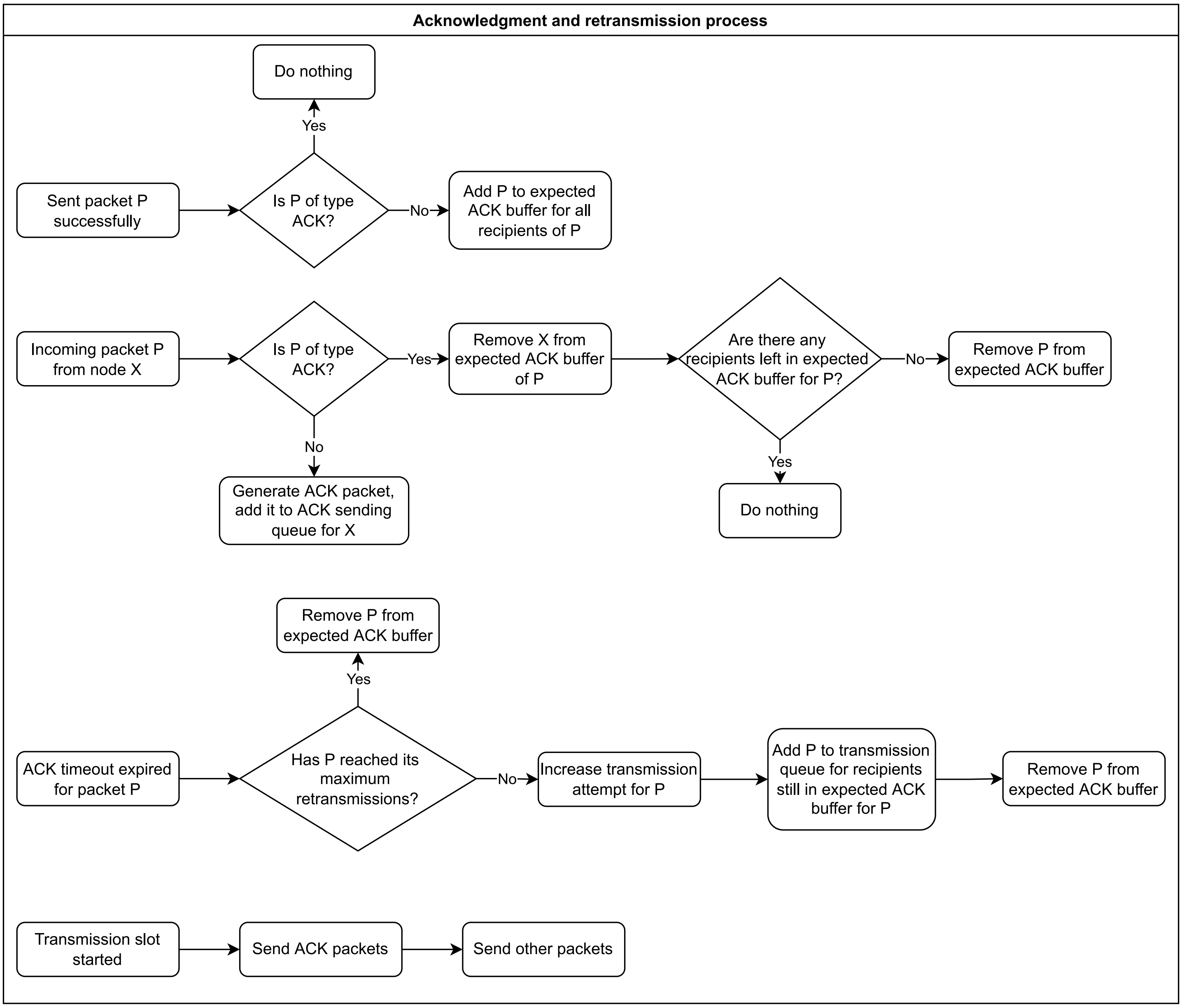
A.2 AllNodesTimeout harvest orchestration strategy


Head Anchor (harvest orchestration strategy: AllNodesTimeout)
Arrival of new hydrophone measurement ID from relay R
Not yet having a measurement with ID from R and ID is not yet processed?
No
Discard measurement
Yes
First measurement with ID from any R?
Yes
Start timeout timer for ID
No
Add measurement to queue
Have a measurement from all R for ID?
No
Do nothing
Yes
Process measurements from ID, cancel timeout for ID
Timeout for ID
Minimum number of measurements reached?
Yes
Process measurements from ID
No
Discard all measurements for ID

Appendix B

Communication process flowcharts

B.1 Acknowledgment and retransmission process


Acknowledgment and retransmission process
Do nothing
Yes
Sent packet P successfully
Is P of type ACK?
No
Add P to expected ACK buffer for all recipients of P
Incoming packet P from node X
Is P of type ACK?
Yes
Remove X from expected ACK buffer of P
Are there any recipients left in expected ACK buffer for P?
No
Remove P from expected ACK buffer
No
Generate ACK packet, add it to ACK sending queue for X
Yes
Do nothing
Remove P from expected ACK buffer
Yes
ACK timeout expired for packet P
Has P reached its maximum retransmissions?
No
Increase transmission attempt for P
Add P to transmission queue for recipients still in expected ACK buffer for P
Remove P from expected ACK buffer
Transmission slot started
Send ACK packets
Send other packets

Appendix C

Specifications acoustic modem used in this study

The specifications listed in the table below are based on the EvoLogics R34D (titanium edition). The whole data sheet for all EvoLogics modems can be found in [10]. This modem has been chosen for its supported operating depth and its frequency band. This frequency band does not interfere with the ULB signal and provides an operating range of several kilometers, as opposed to modems in frequency bands above 100kHz, which have a typical range of a few hundred meters.

Max. operating depth	6000m
Frequency band	18-34 kHz
Max. operating range	3500m
Beam pattern	Horizontally omnidirectional
Power consumption Stand-by	2.5mW
Power consumption Listen mode	5-285 mW
Power consumption Receive mode	0.8W
Power consumption Transmit mode	Up to 65W
Power supply	12V

Appendix D

UWALSim monitors complete structural overview

Monitor type	**Attribute**	**Description**
Battery monitor key: node ID	Battery capacity left	Energy (J) left in the battery
	Estimated time of death	Simulation time (s) that the battery is estimated to run out of resources
Battery transaction monitor key: node ID	Transaction time	Time in simulation (s) of this battery transaction
	Consumption	Operation power consumption (mW)
	Duration	Duration (s) of this operation
	Capacity left	Battery capacity (J) left after this operation
Ping monitor key: ULB transmission ID	Transmission timestamp	Simulation time (s) when the ULB transmitted this ping
	Out of hydrophone misses (including multipath)	Percentage of receivers that did not receive this ping, including multipath arrivals and signals that bounced from surface or bottom areas in the ocean
	Out of hydrophone misses (excluding multipath)	Percentage of receivers that did not receive this ping, excluding multipath arrivals and bounces (signals were always direct line-of-sight)
	Packet decipher errors	Number of times that an acoustic data packet containing this measurement could not be deciphered due to corrupt demodulation and failing FEC
	Busy acoustic receiver drops	Number of times that an acoustic data packet containing this measurement was not received because the receiver was busy with another operation (another incoming packet or it was transmitting a packet itself)
	# arrived measurements on Head Anchor	From all possible received measurements, how many are actually delivered to the head anchor
	Time of data completion on Head Anchor	Simulation time (s) when Head Anchor's Harvest Orchestrator forwarded data for localization.
	Total end2end delay	Total time (s) between Transmission timestamp and finishing location estimation
	Total localization error	Total error (m) between actual and estimated ULB location
	Residual	Total square sum of residuals on elected location estimate
Self positioning monitor	Node ID	ID of the node that runs a self-positioning query
	Positioning type	What strategy is used for self-positioning (GPS / Inertial)
	Time	Simulation time (s) when node queries for self-positioning
	Actual position	Actual position of node upon querying self-position
	Estimated position	Estimated position of self-positioning strategy
	Error	Error (m) in estimated position
Acoustic packet monitor key: packet ID	TX node ID	ID of the node that transmits the packet
	RX node ID	ID of the node that is recipient to the packet
	Data type	What type of packet is transmitted (Measurement / Acknowledgment)
	Attempt	Delivery attempt of packet
	Time tx starts sending	Simulation time (s) when packet is transmitted
	Time rx received	Simulation time (s) when packet is received
	Symbol Error Rate	Rate of how many symbols in this transmission were wrongly demodulated
	Status	The status of the packet (queued, transmitting, transmitted, received, error + reason of error)
RF packet monitor key: packet ID	TX node ID	ID of the node that transmits the packet
	RX node ID	ID of the node that is recipient to the packet
	Data type	What type of packet is transmitted (Measurement / Acknowledgment)
	Time tx starts sending	Simulation time (s) when packet is transmitted
	Time rx received	Simulation time (s) when packet is received
	Status	The status of the packet (queued, transmitting, transmitted, received)

Appendix E

Configuration files in UWALSim

E.1 Main configuration file

config.ini

```
[Simulation]
# Time interval (in ms) that the GUI Window is refreshed
WindowRefresh = 1000
# Time interval (in seconds) that node displacements are recalculated
NodePositionUpdateInterval = 1
# Seed for random number generations
Seed = 10

# Simulation time in minutes
RunTime = 120

[Ocean]
# Depth (meters)
Depth = 6000
# Shipping activity (0-1)
ShippingActivity = 0
# Wind speed (m/s)
WindSpeed = 0

####[MAC protocols]####
[FixedTDMA]
# Time slice length that each node in channel is allowed to transmit (seconds)
TxTimeslotLength = 1
# Time to wait before starting next transmission
# (to let latest transmission's signals propagate further) (seconds)
# (worst-case calculated using 1250m inter-relay distance and
# lowest possible sound speed of 1492 m/s)
GuardTime = 0.84

[DynamicTDMA]
# Multiplier for time slot length, and time slot length of bottom-relay node
Multiplier = 1
# Time to wait before starting next transmission
# (to let latest transmission's signals propagate further) (seconds)
# (worst-case calculated using 1250m inter-relay distance and
# lowest possible sound speed of 1492 m/s)
```

```
GuardTime = 0.84

#######[Devices]#######
[BlackBox]
# Frequency of Box in kHz
Frequency = 37.5
# Periodicity of Black Box in seconds
Period = 10
# Signal transmitter strength (dB re 1 uPa @1m)
TransmitStrength = 160.5
# Maximum range (m)
MaximumRange = 3000
# Directivity
Directivity = None
# Transmitting time in milliseconds
TransmitTime = 10
# Time (s) that black box floats before starting to sink and transmit ULB signals
FloatTime = 1200
# Time (s) after which simulator starts registering ULB ping events
# (e.g. continuing from a certain simulation time)
WaitTime = 0
# Sinking speed of BlackBox (m/s)
SinkSpeed = 1

[RelayLink]
# Relay link distance between relay nodes (m)
RelayDistance = 1250
# Relay link deployment speed (m/s); vertical velocity of relay nodes
DeploymentSpeed = 1.5

[Engine]
# Maximum speed (m/s) an anchor engine can displace in the ocean
MaxSpeed = 2

#######[Self positioning]#######
[GPS]
# GPS error in meters
# X and Y values acquired from Specht, M. (2021). doi:10.1017/S0373463321000485
XErrorMean = -0.042
XErrorStd = 1.067
YErrorMean = 0.031
YErrorStd = 0.634
ZErrorMean = 2
ZErrorStd = 2.5
# Update interval in seconds
UpdateInterval = 1

[Inertial]
# Error on pressure sensor (meters) (uniformly distributed on -depth_error and depth_error)
ZError = 5
# Standard error on relative X movements per meter
```

```
XError = 0.8
# Standard error on relative Y movements per meter
YError = 0.8
# Update interval in seconds
UpdateInterval = 1

####[Propagation models]####
[SimpleTerrestrial]
# Speed of light (m/s)
LightSpeed = 300000000

######[Transceivers]######
[GenericAcousticTransceiver]
# Transceiver characteristics based on EvoLogics R 12/24 data sheet
# Operating frequency in Hertz (currently not in use)
FrequencyLow = 18000
FrequencyHigh = 34000
# Directionality in degrees (currently not in use)
Directionality = 70
# Range in meters
MaximumRange = 3500
# Symbol Rate in bps
SymbolRate = 10000
# Symbol alphabet size for PSK constellation (2=BPSK)
SymbolSize = 2
# Signal to Noise ratio (dB)
SNR = 30
# Number of Error Correction Code bytes per packet
ECC = 4
# Total of transmission attempts
MaxTransmissionAttempts = 2
# Time to wait for an acknowledgment before retransmitting a packet (seconds)
AckTimeout = 4

[SimplifiedRFTransceiver]
# Transceiver characteristics based on Digi XTend 1Watt 900 MHz RF modem
# https://www.digi.com/products/networking/gateways/xtend-900mhz-rf-modems#specifications
# Operating frequency in Hertz
FrequencyLow = 902000000
FrequencyHigh = 928000000
# Directionality in degrees (currently not in use)
Directionality = 360
# Range in meters (assuming outside range with max transmission output of 1W)
MaximumRange = 14000
# Connection bandwidth in kbit/s
Bandwidth = 125

######[Message sizes]######
[MessageSizes]
# Automatically calculate message sizes (1 = ON, 0 = OFF)
AutoSize = 1
```

```
# If AutoSize is off, the following configurations are used
# Sizes in bytes
Header = 34
RelayMeasurement = 32
AnchorMeasurement = 64
CommandMessage = 8
MessageAcknowledgment = 4

######[Strategies]######
[AllNodesTimeout]
# Max. time after receiving first measurement of a transmission to stop waiting
# for arrival of measurements (seconds)
# Recommended to set this value to at least 2 times the total window time
Timeout = 28

[AlwaysListen]
# Time (ms) that it takes to process a measurement after black box time period
ProcessingTime = 100

[TDOALS]
# Sound speed used in calculations (m/s)
SoundSpeed = 1496
# Number of permutations on a measurement (#I)
NrPermutations = 2
# Number of processes used to calculate permutations
NrProcesses = 4
# Residual threshold (%) that triggers new permutation calculations
# If permutations from previous measurements lead to residuals larger than
# the previous residuals by x%, calculate new ones
ResidualThreshold = 1
```

E.2 Energy consumption configuration file

energy.ini

```
[EnergyConsumption]
# Energy consumption values are in mW
#######[Sensors]#######
# Hydrophone
Hydrophone.IDLE = 0
Hydrophone.ACTIVE = 180

#######[Self positioning]#######
# GPS
GPS.IDLE = 0
GPS.ACTIVE = 24
# Inertial
Inertial.IDLE = 0
Inertial.ACTIVE = 1
```

```
######[Actuators]######
# Acoustic pinger
Pinger.IDLE = 1
Pinger.ACTIVE = 20000
# Anchor engine
Engine.HOLD_POSITION = 10000
Engine.MOVING = 100000
RelayLink.DEPLOYING = 5000

########[Nodes]########
# Anchor processor based on avg power consumption RPI 3B
Anchor.cpu.IDLE = 1400
Anchor.cpu.SLEEP = 1
Anchor.cpu.LOW = 2400
Anchor.cpu.ACTIVE = 3700
# Relay processor based on avg power consumption Arduino Mega
Relay.cpu.IDLE = 40
Relay.cpu.SLEEP = 1
Relay.cpu.LOW = 90
Relay.cpu.ACTIVE = 400

####[Communication]####
# Source: EvoLogics Acoustic Modems R 12/24
Transceiver.GeneralAcoustic.IDLE = 2.5
Transceiver.GeneralAcoustic.LISTENING = 285
Transceiver.GeneralAcoustic.RECEIVING = 800
Transceiver.GeneralAcoustic.TRANSMITTING = 65000
# Source: Digi XBee SX 1-Watt 900MHz modem, manual section Power Requirements (page 8-10)
# Assuming 1W transmission power output
# Listening mode here is in this data sheet referred to as Idle (cyclic sleep intervals);
# assuming worst-case
# Idle mode here is considered turning off the device
Transceiver.SimpleRF.IDLE = 0
Transceiver.SimpleRF.LISTENING = 108
Transceiver.SimpleRF.RECEIVING = 270
Transceiver.SimpleRF.TRANSMITTING = 3870

[BatteryCapacity]
# Battery capacity values are in mAH
Anchor.battery.capacity = 10000
Anchor.battery.initial = 10000
Relay.battery.capacity = 500
Relay.battery.initial = 500
BlackBox.battery.capacity = 12400
BlackBox.battery.initial = 12400
```

Appendix F

Experiment configurations

For the purpose of reproducing simulation results shown in section 5, the used configuration values are listed in this chapter.

F.1 Ocean current profile

Utilized ocean currents for simulation can be found here. Note that current values are linearly spaced inbetween ocean depths.

Depth (km)	Current X (m/s)	Current Y (m/s)
0	-0.3	0.4
1	-0.3	0.4
2	-0.2	0.2
3	-0.1	0.1
4	0.05	0.05
5	-0.05	-0.08
6	-0.041	-0.05

F.2 Configuration values

All base configuration is based on the configuration file listed in Appendix E.1. This section lists the configuration in each scenario and variation that overrules the base configuration.

Variations

Moderate self-positioning

This variation has no changes with respect to the base configuration found in Appendix E.1.

Perfect self-positioning

```
[GPS]
XErrorMean = 0
XErrorStd = 0
YErrorMean = 0
YErrorStd = 0
```

```
ZErrorMean = 0
ZErrorStd = 0

[Inertial]
ZError = 0
XError = 0
YError = 0
```

Bad self-positioning

```
[Inertial]
XError = 10
YError = 10
```

Scenarios

Black box floating for 20 minutes (scenario 1, 3, 5, 7)

This scenario has no changes with respect to the base configuration found in Appendix E.1.

Black box sinking immediately (scenario 2, 4, 6, 8)

```
[BlackBox]
FloatTime = 0
```

Black box inside triangle (scenario 3, 4)

Change in *modules/sim/world/Factory.py:58*

```
b = BlackBox([...], start_pos=(1200, 1200, 0) [...]
```

Black box sinking faster than relay nodes (scenario 5, 6)

```
[BlackBox]
SinkSpeed = 2
```

Constant sound speed (scenario 7, 8)

Change in *modules/propagation_models/Bellhop.py:34*

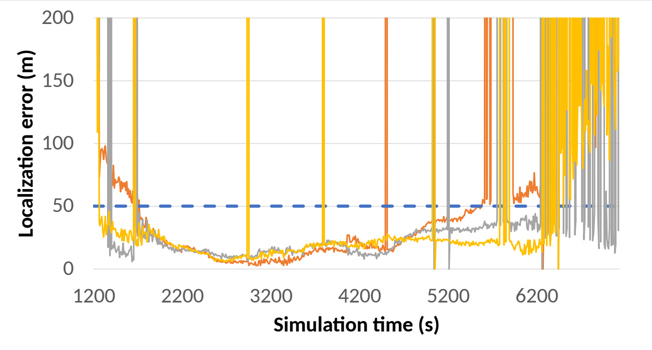
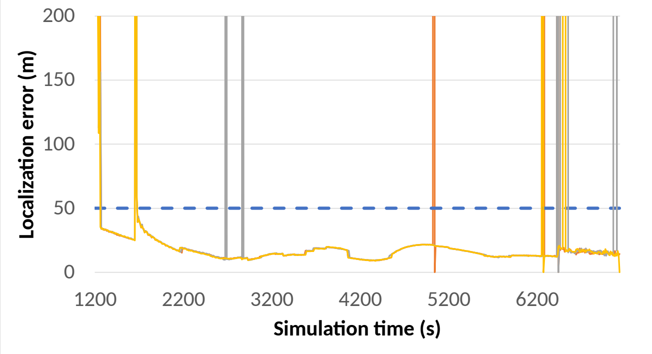
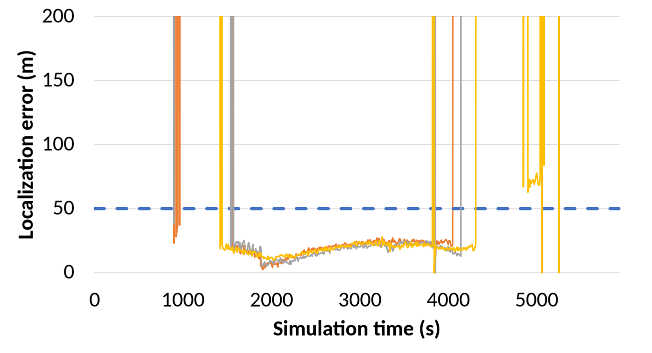
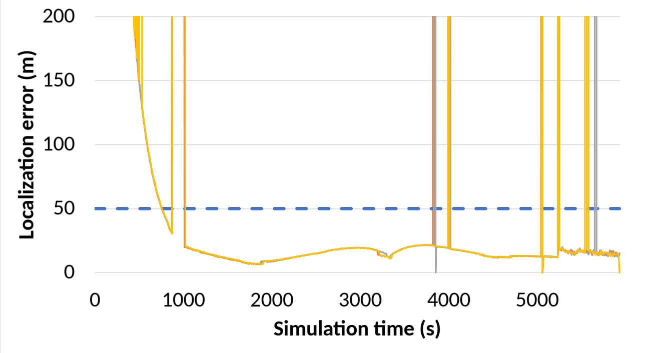
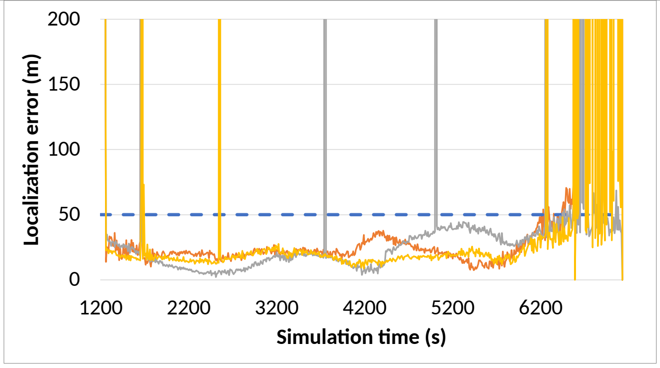
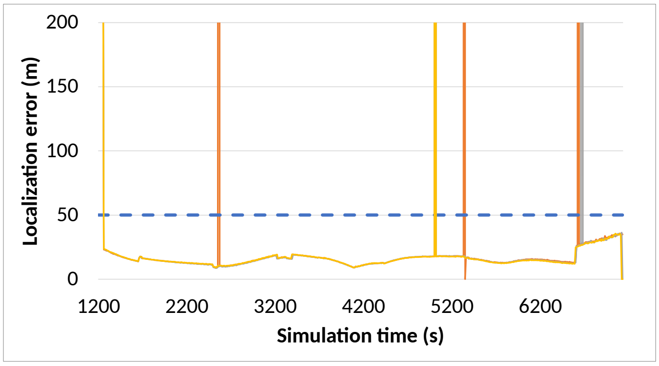
```
‘soundspeed’: 1496,
```

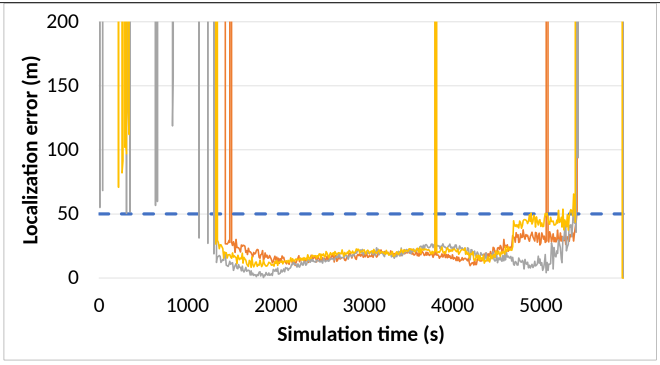
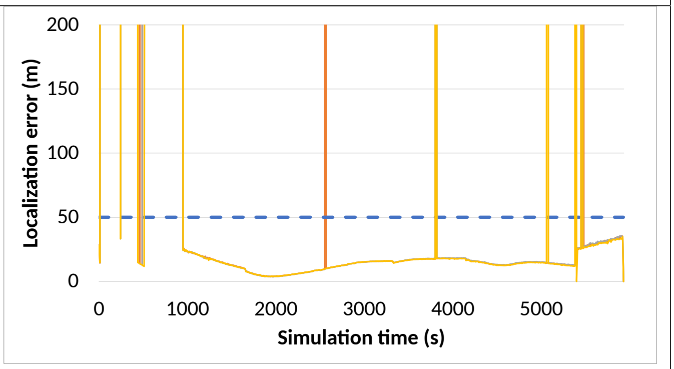
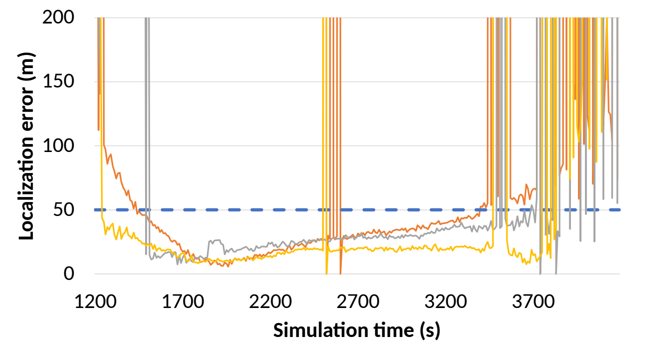
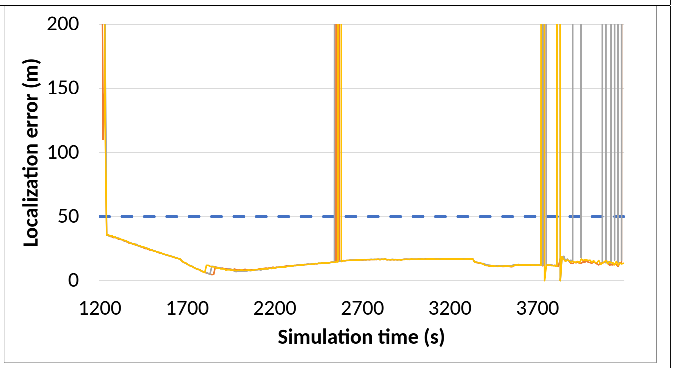
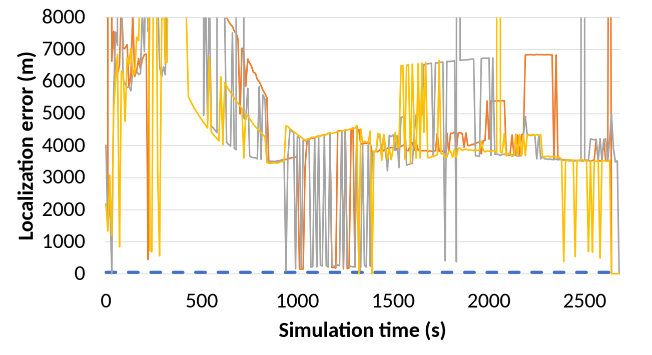
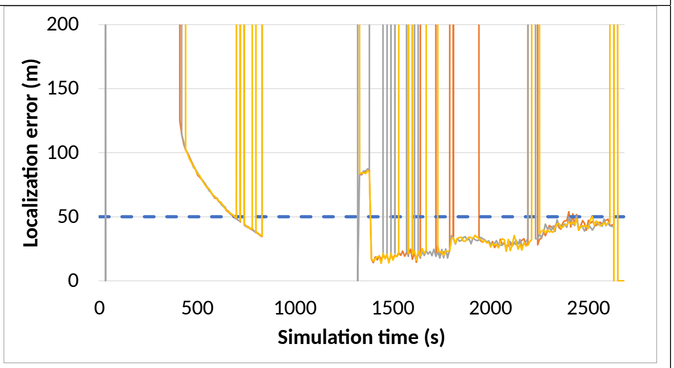
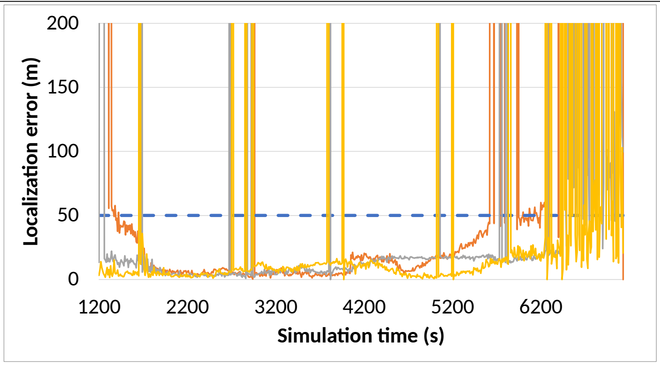
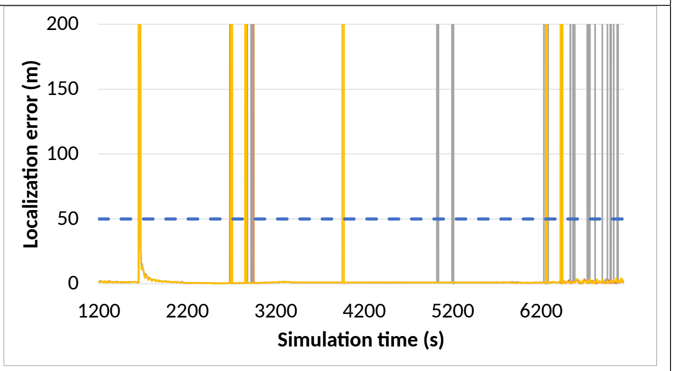
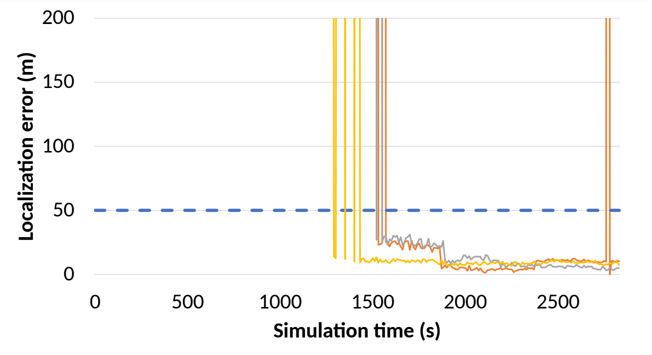
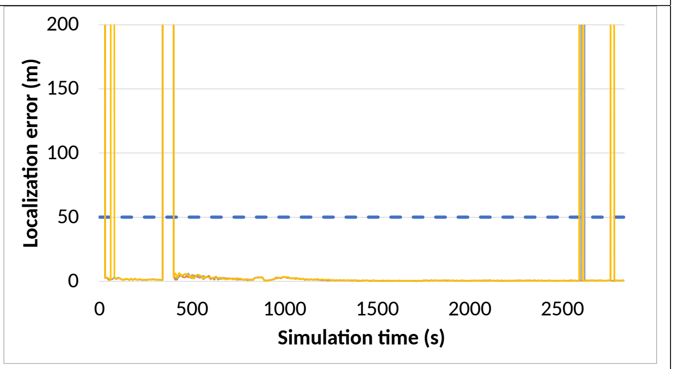
Appendix G

Experimental result figures

G.1 Localization error over time

In this section the results for localization errors are shown following the scenarios described in section 5.1. The 'Bad self-positioning' variant has been omitted since it does not provide useful insight in the localization performance over time. Every graph shows 3 runs with distinct colors, each indicating a seed setting. The dashed blue line indicates the 50-meter error threshold as stated in research question 1 from section 1.2.

Scenario	Moderate SP	Perfect SP
1		
2		
3		

Scenario	Moderate SP	Perfect SP
4		
5		
6		
7		
8	 200 150 100 50 0 Localization error (m) 0 500 1000 1500 2000 2500 Simulation time (s)	 200 150 100 50 0 Localization error (m) 0 500 1000 1500 2000 2500 Simulation time (s)