



SAPIENZA
UNIVERSITÀ DI ROMA

Responsive Navigation, Realistic Trials: Adaptive Autonomous Underwater Vehicles Acoustic Positioning Techniques and Communication-Aware Cyber-Physical Simulation.

Department of Computer, Control and Management Engineering
PhD Program in Engineering in Computer Science (XXXVII cycle)

Luca Iezzi

ID number 1565276

Advisor

Prof.ssa Chiara Petrioli

Co-Advisor

Prof. Giorgio Grisetti

Academic Year 2024/2025

Thesis defended on September 18, 2025

in front of a Board of Examiners composed by:

Prof. Francesco Leotta - Università degli Studi di Roma La Sapienza (chairman)

Prof. Ivan Serina - Università degli Studi di Brescia

Prof.ssa Federica Mandreoli - Università degli Studi di Modena e Reggio Emilia

Prof.ssa Chiara Ghidini - Università di Bolzano

External Reviewers:

Prof. Suman Banerjee - University of Wisconsin–Madison

Prof. Vincenzo Mancuso - Università degli Studi di Palermo

Responsive Navigation, Realistic Trials: Adaptive Autonomous Underwater Vehicles Acoustic Positioning Techniques and Communication-Aware Cyber-Physical Simulation.

PhD thesis. Sapienza University of Rome

© 2025 Luca Iezzi. All rights reserved

This thesis has been typeset by L^AT_EX and the Sapthesis class.

Author's email: iezzi@diag.uniroma1.it

Abstract

The oceans remain the least explored biome on Earth, as the robotic and networking technologies that have empowered land, air, and space completely lose their effectiveness, since radio waves do not penetrate water. As of today, the already constrained acoustic channel is the only mature technology that can sustain the two main challenges of underwater systems: localization and communication. Moreover, testing novel approaches at sea involves costly operations and equipment, and the noisy and time-varying nature of the marine environment severely hinders the ability of performing controlled and rigorous experimental campaigns.

This thesis addresses two closely related obstacles to autonomous underwater exploration: (i) the need for autonomous, accurate localization in non-stationary environmental conditions, and (ii) the absence of realistic, low-cost platforms to test and refine such capabilities before committing to expensive and unpredictable sea trials.

An analytically grounded acoustic localization framework for Autonomous Underwater Vehicles (AUVs), validated in real deployments, is first proposed, exposing how even simple, deterministic models can yield meter-level accuracy. In depth analyses, however, reveal how static noise modeling undermines reliability under time-varying environmental and sensors noise. To address this, the thesis introduces an adaptive localization architecture that is able to tune both process and sensor models, corrects for varying noise conditions, and harmonizes asynchronous acoustic ranges with inertial and complementary data using probabilistic filtering techniques. This approach achieves robust performance, improving localization accuracy to more than 50% with respect to deterministic solutions, without requiring laborious calibration, even in the presence of non-stationary disturbances.

Yet, evaluating complex solutions that aim at precisely characterizing the noise parameters of the system demands testbeds that are able to capture the environmental nuances they mimic. Field trials, though invaluable, are costly and often unsuitable for repeatable, large-scale algorithm development. To fill this gap, this thesis proposes **MAREA**, the first cyber-physical co-simulation framework bridging high-fidelity vehicle dynamics, and realistic underwater communications and sound propagation. By tightly coupling a physics-aware robotic simulator with a modified ns-3 engine that supports multimodal underwater links and stack reconfiguration, **MAREA** provides a realistic "digital ocean" for experimentation. Its physically grounded modeling of hydrodynamics, noise, and synchronization effects bridges the gap between *in silico* accuracy and field uncertainty.

Together, these contributions advance the foundations of underwater autonomy, offering both a framework able to model the nuances typical of real scenarios and methods that adapt to reality.

Contents

List of Figures	vii
List of Tables	xi
List of Abbreviations	xi
1 Introduction	1
1.1 Thesis Roadmap	4
2 Background	5
2.1 Underwater Acoustic Channel Constraints	5
2.2 Unmanned Underwater Vehicle (UUV) Localization Technologies and Architectures	7
2.2.1 Inertial Navigation/Dead-Reckoning and Complementary Sensors	9
2.2.2 Acoustic Ranging	13
2.3 Probabilistic State Estimation and Representation	15
2.3.1 A Bayesian Perspective on State Estimation	15
2.3.2 Extended Kalman Filters	16
2.3.3 Factor Graphs	17
3 Acoustic Localization of Autonomous Underwater Vehicles	21
3.1 Experimental Settings and Scenario	23
3.2 Experimental At-Sea Evaluation of a Baseline EKF-Based Localization Approach	26
3.2.1 System Model	27
3.2.2 Experimental Evaluation	29
3.2.3 Results Discussion	32
3.2.4 Related Work	34
3.3 An Adaptive Extended Kalman Filter for State and Parameter Estimation in AUV Localization	35
3.3.1 Beacon Positions and Noise Statistics Estimation	36
3.3.2 System Model	39
3.3.3 Experimental Evaluation	41
3.3.4 Results Discussion	48
3.3.5 Related Work	48

4	MAREA: a Multimodal Co-Simulation Framework for Marine Net-	53
	worked Agents	
4.1	Related Work	54
4.2	MAREA Framework Design	56
4.2.1	MAREA-CP: Extensions to UUV Simulator	57
4.2.2	MAREA-ns3: Extensions to ns-3	59
4.2.3	Inter-simulator communication and synchronization protocol	61
4.2.4	Modularity and typical usage	63
4.3	Use Case Validation Example	63
4.4	Conclusions	66
5	Conclusions and Future Works	69
5.1	Summary of Contributions	69
5.2	Future Directions	70
5.3	Final remarks	70
A	A Practical MAREA Example on a Factor Graph Localization Pipeline	73
A.1	System Model	73
A.2	Simulation Experiments	75
	Bibliography	79

List of Figures

1.1	Typical IoUT infrastructure with static and mobile underwater assets, ground and surface stations. <i>Source</i> : [1].	2
1.2	Main categories of UUVs. (a) Remotely Operated Vehicle (ROV). (b) Autonomous Surface Vehicle (ASV). (c) Autonomous Underwater Vehicle (AUV).	2
2.1	Absorption coefficient, $10 \log a(f)$ in dB/km	6
2.2	(a) (Approximated) Wenz curves, describing relative levels of marine ambient noise from weather, wind, shipping and thermal noise. (b) SNR levels wrt frequency. 0 dB is the bare minimum to distinguish a received signal from noise, ~ 20 dB is usually enough to be considered for reliable communication.	6
2.3	Sound speed profile with respect to water depth.	7
2.4	Schematic view of the main underwater localization techniques grouped by their working principles.	8
2.5	Example of a DVL mounted on the bottom part of an UUV hull. . .	10
3.1	(a) Camera capture of the shipwreck. (b) Wreck mapping (right side of the scan) through a Side Scan Sonar.	23
3.2	Aerial view of the beacon deployment.	24
3.3	The Zeno AUV.	24
3.4	(a) WSense acoustic modem integrated in the hull of the Zeno AUV. (b) The Zeno AUV during navigation in the waters of Ponza.	25
3.5	Mission trajectory measured by the onboard GPS and beacons positions.	26
3.6	Acoustic ranges measured by the AUV's modem during the mission.	27
3.7	2D Trajectory estimated by the proposed method.	30
3.8	2D velocity estimated by the proposed method.	31
3.9	Diagnostic plots for EKF accuracy and consistency. (a) Histogram with Rayleigh fit; (b) Q-Q plot vs. Rayleigh; (c) Empirical CDF of ε_k ; (d) NIS consistency check; (e) Rolling RMSE and 95 %.	33
3.10	Position estimation of three deployed beacons (orange rhomboids) in a simulated scenario. During the AUV route (blue/cyan dots) the intersections of the circumferences described by the range towards each of the three beacons allow to determine their unique position. .	38

3.11	Estimated trajectory after beacon localization (black solid) vs. estimated trajectory with biased beliefs about the beacon locations (light blue).	43
3.12	Comparison between the artificially induced Gaussian noise perturbing the yaw measures (green) and the noise estimated by our method (orange).	44
3.13	Comparison between the artificially induced half-normal noise perturbing the range measures (green) and the residuals estimated by the presented method (orange) on each beacon (top to bottom). . .	45
3.14	2D Trajectory estimated by the proposed method and the resulting localization error. The green upside-down triangles show the adjust beacon coordinates estimated during the GPS-aided phase.	46
3.15	Range residuals estimated by the presented method (orange) on each beacon (top to bottom) in the sea trials.	47
3.16	(In radians - unwrapped) Roll, Pitch and Yaw angles estimated during the mission.	47
3.17	(In degrees - polar coordinates) Roll, Pitch and Yaw angles estimated during the mission.	48
3.18	Diagnostic plots for the adaptive EKF accuracy and consistency. (a) Histogram with Rayleigh fit; (b) Q-Q plot vs. Rayleigh; (c) Empirical CDF of ε_k ; (d) NIS consistency check; (e) Rolling RMSE and 95 %. .	49
4.1	Software architecture of MAREA simulator showing high-level perspectives of MAREA-CP and MAREA-ns3, and their internal modules.	57
4.2	Mapping between Robot Model or Model instances in GAZEBO and Node instances in ns-3, as well as exchanges between components of MAREA simulator.	58
4.3	Support for multiple networking stacks in MAREA-ns3; one or more NET, MAC, or PHY instances can compose a node's networking stack, transparently used through the ns-3 Socket API, and enabled for self-configuration via layer-specific configurators.	61
4.4	Inter-simulator communication protocol through sockets.	62
4.5	Explanation of the ϵ_{range} metric in the ranging use case. τ_{prop} is the propagation delay for a communication initiating at time t ; w.l.o.g., the communication is received after k GT updates plus desynchronization delay τ'	64
4.6	Average ranging error, varying GT update frequency τ_{upd}	65
4.7	Average T_{Wall} overhead w.r.t. T_{ROS} , varying number N_a of AUVs and GT update frequency τ_{upd}	66
4.8	Desynchronization delay τ' , varying GT update frequency τ_{upd}	67
A.1	Simulated environment of the proposed scenario. Four AUVs (with corresponding four differently colored trajectories and real time position and orientation estimates) in a scenario with four deployed beacons (red boxes in the background).	76
A.2	Four AUVs' Ground Truth (solid trajectories) versus the estimate given by the proposed approach (dashed trajectories).	77

A.3 Localization error (RMSE, MAE) of the proposed approach for each deployed AUV.	78
---	----

List of Tables

2.1	Range versus data-rate characteristics of common underwater communication technologies.	5
2.2	Key characteristics of common sensors used on board underwater vehicles to complement Inertial Navigation / Dead-Reckoning.	12
3.1	Technical specifications of the WNode's acoustic modem.	24
3.2	Summary of Ponza field campaign settings.	28
3.3	Noise statistics used in the present EKF formulation.	30
3.4	Initial EKF state and covariance parameters used in simulations and field experiments.	42
3.5	Estimated σ_r^2 versus injected $\sigma_{noise,r}^2$ range noise in the simulated scenario.	45
4.1	Simulation parameters.	64
A.1	System and noise parameters used in the simulated scenario.	77

List of Abbreviations

AoA Angle of Arrival. 13

AUV Autonomous Underwater Vehicle. 4, 7, 21, 23–29, 34–37, 39–46, 51, 63, 70, 73, 75–77

CDF Cumulative Distribution Function. 32, 46

COG Course-Over-Ground. 36, 40, 41

CPS Cyber-Physical System. 53, 54

DCM Direction Cosine Matrix. 39

DGPS Differential GPS. 34

DoF Degrees-of-Freedom. 25

DVL Doppler Velocity Log. 9, 25, 31, 34, 35, 50, 51

ECEF Earth-Centered Earth-Fixed. 12, 13

EKF Extended Kalman Filter. 4, 15, 16, 19, 23, 26, 29–32, 34–36, 38–42, 44–46, 48, 50

ENU East-North-Up. 12

GNSS Global Navigation Satellite System. 2, 9

GPS Global Positioning System. 21, 23–25, 29, 36, 37, 40–45

IMU Inertial Measuring Unit. 22, 23, 25, 34–36, 39, 41, 50, 75, 76

INS Inertial Navigation System. 9

IoUT Internet of Underwater Things. 1, 3

iSAM Incremental Smoothing and Mapping. 18, 19, 73, 74

KF Kalman Filter. 16, 36

LBL Long Baseline. 14, 22, 26, 34–36, 63

- MAP** maximum-a-posteriori. 17, 18
- NED** North-East-Down. 10–13, 39, 40
- NIS** Normalised Innovation Squared. 32, 46, 48
- PDF** Probability Density Function. 32
- PDoA** Phase Difference of Arrival. 13
- Q-Q** Quantile-Quantile. 32, 47, 48
- RF** Radio Frequencies. 2, 5
- RMSE** Root Mean Square Error. 26, 29, 30, 32, 36, 41, 44, 46, 47, 50, 77
- SBL** Short Baseline. 22
- SLAM** Simultaneous Localization And Mapping. 8
- SNR** Signal-to-Noise Ratio. 6
- SSP** Sound Speed Profile. 7, 44
- TDoA** Time Difference of Arrival. 13
- ToA** Time of Arrival. 13
- ToF** Time-of-Flight. 6–8, 13, 14, 44
- UAV** Unmanned Aerial Vehicle. 1, 55
- UKF** Unscented Kalman Filter. 34
- USBL** Ultra-Short Baseline. 22
- UUV** Unmanned Underwater Vehicle. v, 4, 7, 9, 11–15

Chapter 1

Introduction

The undersea domain comprises more than 70% of the Earth’s surface and represents a largely unexplored frontier of paramount importance. The oceans are critical to climate regulation, marine biodiversity and global trade, and underpin fast-growing Blue Economy [2] sectors including environmental monitoring, aquaculture, renewable energy and maritime security. In fact, the world’s oceans regulate climate, host the planet’s richest biodiversity and carry more than 90% of global trade. Yet the oceans remain largely under-explored: only 25% of the seafloor has been mapped to 100 m resolution or better, leaving three quarters of Earth’s largest habitat effectively unknown to us.

Unlocking this “blue frontier” demands an infrastructure that can sense, act and communicate reliably underwater. The smart interconnected infrastructures that enable applications in the underwater domain are commonly referred to as Internet of Underwater Things (IoUT), and are composed by networked assets that are capable of wirelessly transmitting and relaying sensed data and instructions among themselves and to external assets such as surface vessels or on-shore terrestrial stations. A typical IoUT infrastructure consists in interconnected smart objects of heterogeneous nature: from surface nodes (e.g., smart buoys or vessels), submerged mobile and fixed nodes, and sometimes even aerial or terrestrial assets such as Unmanned Aerial Vehicles (UAVs) and onshore monitor centers, as depicted in Figure 1.1.

Particularly, Unmanned Underwater Vehicles (UUV) (Figure 1.2) have emerged as indispensable tools in this context, enabling safer, cheaper, and more extensive exploration and monitoring of large geographical areas, decreasing the costs of tethered infrastructures, reducing human intervention in hazardous areas, and extending network coverage thanks to their mobility. Over the years, an increasing number of fundamental applications for marine geoscience and ocean studies [3–5], inspection and maintenance of submerged infrastructures and search and rescue missions [6], and for harbor safety and surveillance [7, 8] have benefited from the use of autonomous vehicles, among many others.

The ever-expanding exploration and exploitation of the marine environment demands the support of solutions on par with those available on earth, to ease swift real-time monitoring and enable context-awareness. Realizing this vision, however, depends on several factors that are highly inter-dependent.

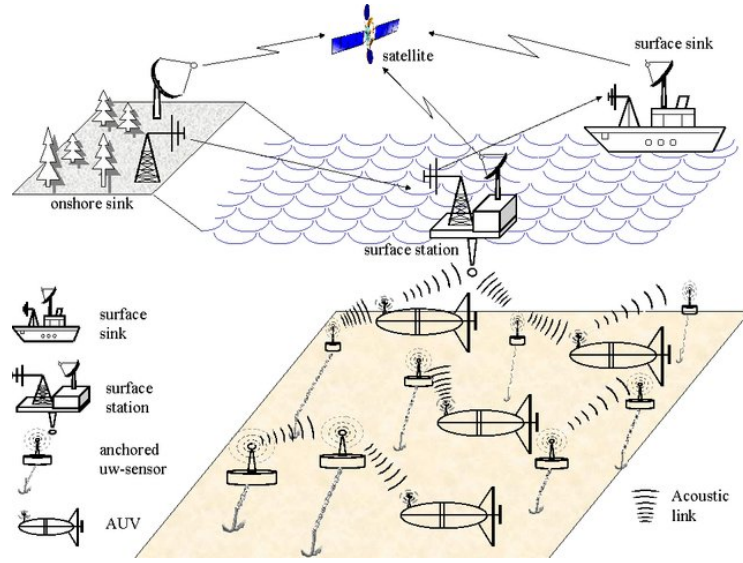


Figure 1.1. Typical IoUT infrastructure with static and mobile underwater assets, ground and surface stations. *Source:* [1].

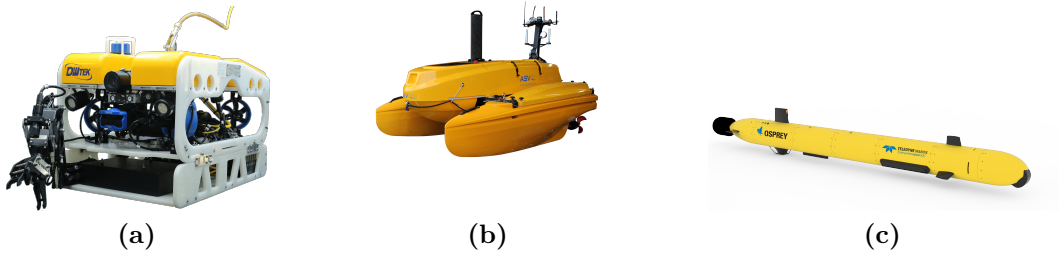


Figure 1.2. Main categories of UUVs. (a) Remotely Operated Vehicle (ROV). (b) Autonomous Surface Vehicle (ASV). (c) Autonomous Underwater Vehicle (AUV).

Accurate real-time localization is the first prerequisite for almost any underwater system, from georeferencing sensor data to coordinating surveys with multiple robots and ensuring collision avoidance. However, unlike terrestrial and aerial positioning, which both relies on Radio Frequencies (RF) signals, vast bandwidth, and line-of-sight links to dense Global Navigation Satellite System (GNSS) or cellular infrastructure, the inability of RFs to penetrate the water medium prevents their use.

While an established approach for ground robots and drones, also vision-based localization incurs in hard limitations. In fact, visible light absorption and scattering underwater make the maximum usable visual range only around 5-10 meters even in good conditions, making cameras only suitable in enclosed environments or where close inspection is involved.

Thus, most localization solutions rely on the same low bandwidth acoustic channel that carries command-and-control traffic, whose bandwidth is limited to hundreds of bits per second and propagation speed is $\sim 1,500$ m/s (five orders of magnitude slower than light). Localization and communication thus compete for

the same physical resources.

The underwater environment is also characterized by a high non-stationarity: its ever-changing temperature-salinity profile, high turbidity, and flow-induced dynamics make the medium time-varying and noisy, degrading both communications and the fidelity of cameras, sonar, and navigation sensors.

Additionally, deploying, maintaining and scaling IoUT infrastructures still impose significant capital and ship-day costs, in the order of thousands to hundred-thousands euros per day, coupled with logistical constraints such as ship time, weather windows, and regulatory permits.

It is thus clear that the non-trivial combined effects of the physical and operational constraints demand a focus on two principal aspects.

- Effective AUV navigation frameworks must integrate adaptive estimation schemes, fusing, for example, inertial, and acoustic data that explicitly account for the stochastic structure of the water column, while remaining robust to the unreliable communication medium.
- The ability of modeling the environment to jointly take into account vehicle dynamics, hydrodynamic forces, bathymetry, and the stochastic acoustic channel is an essential requisite. This way, control laws, sensor fusion strategies, and network protocols can be tested in realistic operational scenarios long before incurring in the costs and impracticalities of setting full-scale sea trials. The use of a “digital ocean” model can mitigate the risks of costly field campaigns, but also accelerates prototyping of new localization methods under repeatable and representative conditions.

In response to the introduced challenges, this thesis proposes three principal contributions:

1. a deterministic baseline acoustic localization pipeline, validated in sea trials with analytically derived performance bounds;
2. an adaptive localization scheme that self-tunes to non-stationary conditions and is able to achieve meter level accuracy in sea trials without complex calibration procedures;
3. MAREA, the first modular co-simulation underwater framework that unifies vehicle dynamics, hydrodynamic forces, seabed bathymetry and a time-varying stochastic acoustic channel;

Collectively, these advances represent a step forward in augmenting underwater systems’ autonomy and situation awareness, lowering the cost of experimentation, shortening the design cycle for future IoUT services and making the oceans more accessible.

The rest of this Chapter proceeds with a roadmap of the thesis structure.

1.1 Thesis Roadmap

The remainder of this thesis is organized as follows.

Chapter 2 introduces the reader to the theoretical background behind the contributions proposed in this manuscript. **Section 2.1** describes the underwater acoustic channel's characteristics, highlighting the physical properties of water that affect acoustic communication and its main sources of error. **Section 2.2** explores the fundamentals of underwater UUV localization, with a focus on the enabling technologies and the acoustic properties that can be leveraged to localize assets in the underwater environment. **Section 2.3** delves into the main Probabilistic Estimation Frameworks used in the proposed localization approaches. The Chapter reviews the *Bayesian* interpretation of the State Estimation problem (**Section 2.3**) and proceeds in presenting the theoretical background of the frameworks at the foundation of the presented localization approaches.

Chapter 3 presents two acoustic-aided localization frameworks for low-cost Autonomous Underwater Vehicles (AUVs) in detail. After reviewing the scenario and settings of the sea trials in which the experiments have been carried out (in **Section 3.1**), the Chapter delves into the system models and experimental evaluation of the proposed approaches. In **Section 3.2** a "baseline" Extended Kalman Filter (EKF)-based approach with time-invariant noise statistics and parameters is presented and evaluated in field tests at sea, achieving meter-level localization accuracy. After reviewing the state of the art in similar settings, this section concludes with a detailed analysis of the noise-induced nuisances that justify the implementation of adaptive approaches that can tackle time-varying noise conditions. In a similar fashion, **Section 3.3** builds on the "baseline" filter analysis and provides the system model and at-sea evaluation of an adaptive localization method that is able to levigate the noise-induced effects observed in the previous solution.

Chapter 4 introduces MAREA, the first co-simulation framework that seamlessly combines realistic underwater physics with a detailed multimodal networking protocol stack. After reviewing the state of the art on cyber-physical simulation and communication in the ground, air and underwater domains, the design on the simulation framework is presented and evaluated in challenging benchmarking tests that reproduce real scenarios where the joint physical modeling and acoustic communication are paramount.

Chapter 5 draws the conclusions and prospects future research stemming from the work presented in this manuscript.

Chapter 2

Background

2.1 Underwater Acoustic Channel Constraints

The limitations of RF and optical-based technologies make sound waves the *de facto* standard for most applications in the underwater domain. The metrics shown in Table 2.1, derived from an analysis by Campagnaro et al. in [9], make the reason clear. In fact, while optical communication is employed in *ad hoc* scenarios where communicating assets are close to each other (e.g., data mule applications with large amounts of data to offload), RFs exhibit no practical property underwater in the current state of research.

Technology	Range (m)	Data rate
Radio-frequency	0.05	100 Mbps
	1	1 Mbps
	8	2 kbps
Optical	10	3 Gbps
	80	10 Mbps
	120	1 Mbps
Acoustic	200	100 kbps
	10 000	1 kbps
	30 000	150 bps

Table 2.1. Range versus data-rate characteristics of common underwater communication technologies.

However, because of the characteristics and physical constraints of acoustic propagation [10], enforcing underwater communication is far from straightforward, as there are many factors that work against establishing a reliable link between the networked assets, that are described in the following.

Attenuation and Noise. Acoustic waves that propagate underwater, experience absorption loss (depending on the frequency, as can be seen in Figure 2.1) and spreading loss (depending on the distance). These can be jointly defined as *Path Loss*, given by:

$$A(l, f) = (l/l_r)^k a(f)^{l-l_r}$$

where f is the signal frequency and l is the transmission distance, taken in reference to some l_r . The path loss exponent k models the spreading loss, and it is an empirical value usually between 1 and 2 (for cylindrical and spherical spreading, respectively).

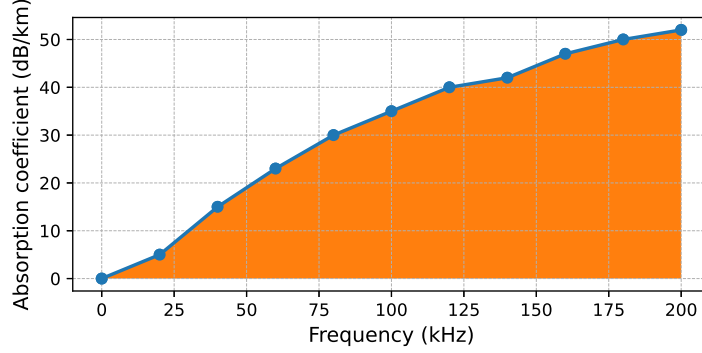


Figure 2.1. Absorption coefficient, $10 \log a(f)$ in dB/km.

While the attenuation grows with frequency, the noise spectrum has the opposite tendency, decaying with increasing frequency (Figure 2.2a): this results in a Signal-to-Noise Ratio (SNR) that varies over the signal bandwidth. Figure 2.2b shows how the acoustic bandwidth depends on the transmission distance l : for example, the bandwidth available to transmit at a distance of 100 km is only ~ 2 kHz, while it increases at shorter distances.

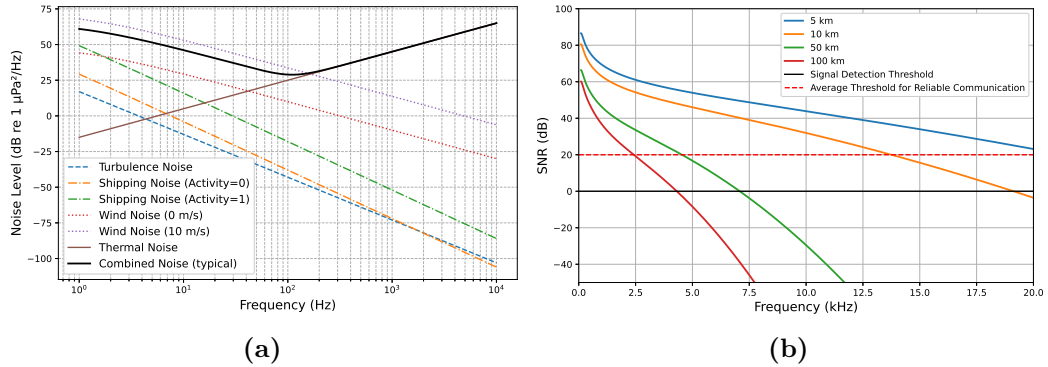


Figure 2.2. (a) (Approximated) Wenz curves, describing relative levels of marine ambient noise from weather, wind, shipping and thermal noise. (b) SNR levels wrt frequency. 0 dB is the bare minimum to distinguish a received signal from noise, ~ 20 dB is usually enough to be considered for reliable communication.

The strong frequency-range coupling forces a severe trade-off between data rate and coverage. Communication-wise, this implies modems being configured toward a narrow-band, power-efficient schemes and aggressive forward-error correction. For localization, a low SNR degrades range accuracy and, at long ranges, the absorption-induced frequency selectivity severely impact Time-of-Flight (ToF) measurements.

Multi-path propagation. Multi-path formation stems from two different phenomena: sound reflection at the water surface/bottom and at any object along the path of the acoustic waves and from sound refraction in water. The latter is a consequence of the Sound Speed Profile (SSP) (Figure 2.3), which varies depending on several factors such as water temperature, salinity and pressure.

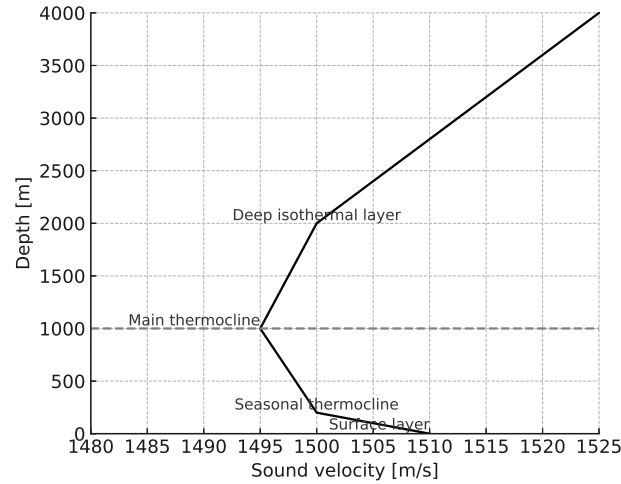


Figure 2.3. Sound speed profile with respect to water depth.

In navigation, the same echoes create multiple and delayed ToF hypotheses, yielding biased ranges unless the first-path component can be isolated.

Time variability and Doppler effect. Short-term changes in the propagation medium conditions or due to the motion of the transmitter/receiver cause frequency spreading and shifting (*Doppler effect*) whose magnitude is proportional to the ratio $a = v/c$, where v is the object speed and c is the sound speed in water. Hence, since c is several orders of magnitude lower than the speed of electromagnetic waves in the air (where frequency shifting can be negligible), motion-induced Doppler effect can be extreme, especially for assets that undergo voluntary manouvers (e.g., AUVs) or involuntary motion (e.g., tides and waves). Relative speeds of only a few knots translate into Doppler shifts of several tens of hertz at typical carrier frequencies (10–30 kHz).

In localization, Doppler manifests as a range-rate error: for example, if not compensated by the acoustic modem, a 2-knot vehicle speed introduces a ~ 0.9 m bias per 100 ms two-way travel time at 1500 m/s sound speed.

Hence, the impairments discussed above do not impact acoustic links in isolation; they jointly characterize the environment within which UUVs must communicate and localize themselves.

2.2 UUV Localization Technologies and Architectures

The problem of localizing underwater vehicles has been investigated for decades, in military, civilian and commercial applications [11] and draws on a diverse ecosystem

of sensors and estimation strategies (Figure 2.4):

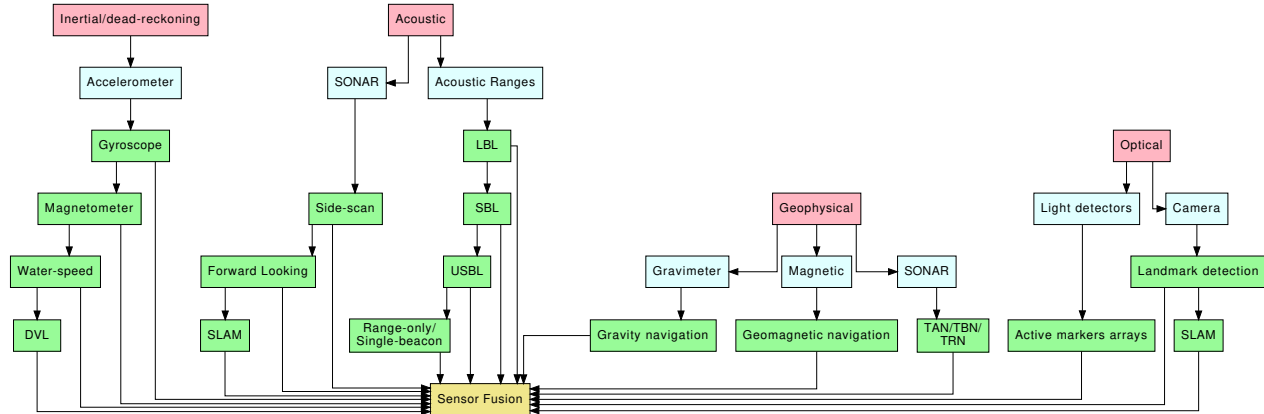


Figure 2.4. Schematic view of the main underwater localization techniques grouped by their working principles.

- **Inertial/Dead Reckoning** sensors rely on the propagation of the vehicle pose by integrating body-frame accelerations and angular rates, possibly bounded by velocity or speed-through-water cues;
- **Acoustic**-based sensors allow absolute or relative positioning with respect to external transponders or natural reflectors via ToF or phase-based ranging;
- **Optical** sensors enable vision-based methods, exploiting the ambient or artificial light for landmark detection, Simultaneous Localization And Mapping (SLAM) or marker-based tracking;
- **Geophysical** sensors exploit environmental fields (e.g., gravity or magnetic field) as natural beacons to achieve terrain-referenced navigation.

Localization techniques that need minimal to no infrastructure in the underwater realm rely on specific environmental features to work properly, and because of this are generally used in *ad-hoc* context. For example, vision-based techniques [12] suffer from turbid and light-scarce waters, and are therefore usually employed in applications where the vehicle is able to survey artificial or natural markers from a few meters (e.g., pipeline inspection).

Sonar-based methods overcome the range problem, but still need a feature-rich terrain to be able to perform loop closure [13] and are usually employed in applications where the seabed presents heterogeneous texture [14] of distinguishable objects (e.g., a shipwreck).

Localization schemes relying on externally-deployed acoustic transponders or platforms, instead, can be employed virtually on any scenario, their only limitations being those related to the sound propagation in water (see Section 2.1). Although these methods were generally considered more cumbersome because of the economic and time expenses required in deploying the infrastructure, the perception is rapidly shifting. In fact, the increasing pervasiveness of large-scale oceans-observing

programmes now blanket many coastal and deep regions with permanent or semi-permanent deployments. Such infrastructures comprise assets that can live weeks to months on a single battery charge, allowing the localization-aiding information to be piggy-backed on the acoustic beacons, relay buoys, and cooperative vehicles that are already deployed. For example, the land and air domains already have a rich ecosystem where global (e.g., GNSS) or local (e.g., Wi-Fi Access Points) scale infrastructure can anchor a navigation solution.

The rest of this section focuses on the foundations of the localization schemes that are of interest in this thesis.

In Section 2.2.1, Inertial Navigation and Dead Reckoning are briefly reviewed along the typical sensors that enable them on UUVs.

Section 2.2.2 delves into the acoustic ranging mechanism for UUV localization.

2.2.1 Inertial Navigation/Dead-Reckoning and Complementary Sensors

Dead-Reckoning refers to the recursive integration of measured vehicle motion in order to infer its current state. For a strap-down Inertial Navigation System (INS), this results in the classical kinematic chain:

$$\begin{aligned}\mathbf{v}(t) &= \mathbf{v}_0 + \int_0^t (\mathbf{f}_b(\tau) - \mathbf{g}(\tau)) d\tau, \\ \mathbf{p}(t) &= \mathbf{p}_0 + \int_0^t \mathbf{v}(\tau) d\tau.\end{aligned}$$

where $\mathbf{f}_b$ is specific force in the body frame measured by a **the tri-axial accelerometer** and $\mathbf{g}$ is gravity projected through attitude estimated through a **tri-axial gyroscope**. The double integration causes an error growth proportional to t^2 , known as *drift*. In typical underwater vehicles, this drift can be bounded using measures from other on-board sensors. The main ones, that are used in the works presented in this manuscript, are reviewed in the following.

Doppler Velocity Logs (DVLs) recover the full 3D velocity over ground of a vehicle by pulsing four narrow-beam oblique acoustic pings toward the seabed (see Figure 2.5) and analyzing the Doppler shift of the returning signal. Let $f_0, f_{r,i}$ be the transmit frequency (Hz) and the return measured frequency on beam i , respectively. Then, the Doppler shift on one beam Δf_i can be used to compute the radial (line of sight) velocity on beam i $v_{r,i}$ as:

$$\begin{aligned}\Delta f_i &= f_{r,i} - f_0 \\ v_{r,i} &= \frac{c \Delta f_i}{2 f_0},\end{aligned}$$

where c is the speed of sound.

Stacking the radial velocities into $\mathbf{v}_r$ and the beam direction cosines into $\mathbf{B}$ (whose rows are the unit direction vector of a beam expressed in the vehicle body

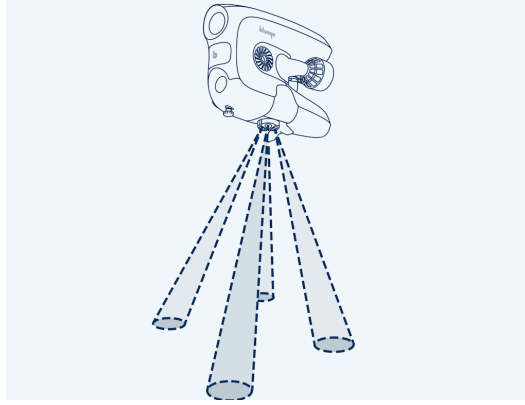


Figure 2.5. Example of a DVL mounted on the bottom part of an UUV hull.

frame b) as:

$$\mathbf{v}_r = \begin{bmatrix} v_{r,1} \\ v_{r,2} \\ \vdots \\ v_{r,N} \end{bmatrix}, \quad \mathbf{v}_b = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{1x} & b_{1y} & b_{1z} \\ b_{2x} & b_{2y} & b_{2z} \\ \vdots & \vdots & \vdots \\ b_{Nx} & b_{Ny} & b_{Nz} \end{bmatrix},$$

yields the linear measurement equation:

$$\mathbf{v}_r = \mathbf{B} \mathbf{v}_b$$

which, for four beams, can be solved using a least-squares estimate:

$$\mathbf{v}_b = (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{v}_r.$$

3-axis Magnetometers measure the local magnetic field vector in the sensor's body frame. By compensating for distortions and for tilt (roll/pitch), it can recover the heading (yaw) relative to magnetic north. Let the rotation matrix from *navigation frame* (NED) to *body frame* be $\mathbf{R}_n^b$. In North-East-Down (NED) coordinates, the Earth's magnetic field (assuming no local anomalies and a pre-calibration phase) can be modeled as:

$$\mathbf{H}_{ned} = \begin{bmatrix} H \cos(D) \\ H \sin(D) \\ D_z \end{bmatrix}, \quad D_z = H \tan(I),$$

where:

- H is the horizontal field intensity,
- D is the declination (angle East of geographic north),
- I is the inclination (angle below the horizontal), and
- D_z is the vertical component in NED (downwards positive).

The expected field in the body frame is then

$$\mathbf{H}_b = \mathbf{R}_n^b \mathbf{H}_{\text{ned}} = \underbrace{\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}}_{\text{calibrated measurement}}.$$

To find the horizontal components in the body frame (compensated for roll ϕ and pitch θ), the roll and pitch are factored out through the the partial rotation matrix:

$$\mathbf{R}_{\text{tilt}} = \mathbf{R}_b^n(\phi, \theta, 0) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ \sin \phi \sin \theta & \cos \phi & \sin \phi \cos \theta \\ \cos \phi \sin \theta & -\sin \phi & \cos \phi \cos \theta \end{bmatrix},$$

where $\mathbf{R}_b^n(\phi, \theta, 0)$ is the rotation from body to NED using roll ϕ , pitch θ , and zero yaw. Applying this to $\mathbf{m}$ yields:

$$\begin{bmatrix} M'_x \\ M'_y \\ M'_z \end{bmatrix} = \mathbf{R}_b^n(\phi, \theta, 0) \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}. \quad (2.1)$$

After tilt compensation, the heading (yaw) ψ is simply:

$$\psi = \text{atan2}(-M'_y, M'_x), \quad (2.2)$$

where $\text{atan2}(y, x)$ returns the angle between the positive x -axis and the point (x, y) , in the range $(-\pi, \pi]$.

Pressure Sensors measure the hydrostatic pressure exerted by the water column above them and relate it to the vehicle depth. Typically, a pressure sensor outputs an absolute pressure, which includes both atmospheric pressure at the surface and hydrostatic pressure due to depth. Let:

$$P_{\text{meas}} = P_{\text{atm}} + \Delta P \quad (2.3)$$

$$\Delta P = P_{\text{meas}} - P_{\text{atm}}, \quad (2.4)$$

where, P_{atm} is the atmospheric (surface) pressure (Pa). Assuming constant fluid density ρ , the hydrostatic pressure due to a column of fluid of height $h(m)$ is:

$$\Delta P = \rho g h,$$

with ρ the density of the fluid, $(kg/m)^3$ and g the gravitational acceleration $(m/s)^2$. By subtracting atmospheric pressure, the depth solely due to the water column can be approximated as:

$$h = \frac{P_{\text{meas}} - P_{\text{atm}}}{\rho_{\text{est}}(T) g}$$

$$\rho_{\text{est}}(T) = \rho_0 [1 - \alpha(T - T_0)],$$

where α is the thermal expansion coefficient, ρ_0 is the density at reference temperature T_0 . This is possible because modern pressure sensor also measure temperature, yielding higher accuracy on the depth estimation.

Table 2.2 summarizes the typical characteristics of the sensors presented in this section.

Sensor	Measured quantity	Typical bias ($1\ \sigma$)	Update rate
Accelerometer	Specific force $\mathbf{f}_{x,y,z}$	50–200 μg	100–1000 Hz
Gyroscope	Body rates $\boldsymbol{\omega}_{x,y,z}$	0.01–0.1°/h	100–1000 Hz
Magnetometer	Earth-field vector $\mathbf{B}$	0.1–1 μT	10–100 Hz
Pressure Sensor	Pressure P	0.5–1 dbar	1–10 Hz
Doppler Velocity Log (DVL)	Velocity over ground $\mathbf{v}_{DVL}$	0.2–0.5 % of reading	1–20 Hz

Table 2.2. Key characteristics of common sensors used on board underwater vehicles to complement Inertial Navigation / Dead-Reckoning.

Global Navigation Satellite Systems (GNSS) , even if unavailable when submerged, can still provide support to UUV positioning, in scenarios where the vehicles rapidly emerge for a position fix, or in missions involving segments in surface navigation mode, such as part of the approaches proposed in this manuscript. They provide raw latitude, longitude and ellipsoidal height in the geodetic frame:

$$\mathbf{z}_{LLA}^{GPS} = \begin{bmatrix} \varphi \\ \lambda \\ h \end{bmatrix},$$

where φ and λ are the latitude and longitude, respectively, and h is the height above WGS-84¹ ellipsoid. Comparing this measurement to the UUV position, typically $\in \mathbb{R}^3$ in the state of state estimation frameworks, requires converting it into a NED (used in this example) or East-North-Up (ENU) world frame through a two-step pipeline:

- **Geodetic to Earth-Centered Earth-Fixed (ECEF).** Let a be the semi-major axis and e the first eccentricity of the WGS-84 ellipsoid. The prime vertical radius of curvature is $N = a / \sqrt{1 - e^2 \sin^2 \varphi}$. Cartesian coordinates are:

$$\begin{aligned} x &= (N + h) \cos \varphi \cos \lambda, \\ y &= (N + h) \cos \varphi \sin \lambda, \\ z &= ((1 - e^2)N + h) \sin \varphi, \end{aligned} \quad \mathbf{r}^{ECEF} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

¹World Geodetic System 1984, is a global geographic coordinate system used for defining positions on the Earth's surface.

- **ECEF to local NED.** Given a reference origin $(\varphi_0, \lambda_0, h_0)$ (usually the first GPS fix of the mission or the surveyed launch point), and its ECEF position $\mathbf{r}_{\text{ref}}^{\text{ECEF}}$ computed as above, the displacement in the local NED frame is then:

$$\mathbf{p}_w^f = \mathbf{T}_{\text{ECEF} \rightarrow \text{NED}}(\varphi_0, \lambda_0)(\mathbf{r}^{\text{ECEF}} - \mathbf{r}_{\text{ref}}^{\text{ECEF}}),$$

where the orthonormal transformation matrix is:

$$\mathbf{T}_{\text{ECEF} \rightarrow \text{NED}} = \begin{bmatrix} -\sin \varphi_0 \cos \lambda_0 & -\sin \varphi_0 \sin \lambda_0 & \cos \varphi_0 \\ -\sin \lambda_0 & \cos \lambda_0 & 0 \\ -\cos \varphi_0 \cos \lambda_0 & -\cos \varphi_0 \sin \lambda_0 & -\sin \varphi_0 \end{bmatrix}.$$

2.2.2 Acoustic Ranging

In underwater acoustics, turning measured time or phase-differences into a spatial fix is dependent on the baseline ² geometry of the anchored transponders (beacons) that form the reference for positioning. Algorithms in underwater positioning [15–17] leverage different signal processing techniques to derive spatial information from the geometry of the system components.

Time Difference of Arrival (TDoA)-based methods the system considers the difference in arrival time between pairs of receivers. Each measurement defines a hyperbola (in 2D) or a hyperboloid (in 3D) from whose intersections position can be deduced.

In systems that exploit **Angle of Arrival (AoA)**, a small array of hydrophones, spaced from a known baseline, estimates the detection of the incoming wavefront. Crossing two or more independent detections, or combining a detection with an interval, locates the source.

In methods based on **Phase Difference of Arrival (PDoA)** the phase difference between two signals received by a transmitter from two or more receivers is measured, allowing the angle of arrival and position of the transmitter to be determined;

This section, however, focuses on methods that base their functioning purely on the **Time of Arrival (ToA)** of the signal exchanged between a target vehicle and one or more deployed acoustic beacons. Such techniques are the foundation of the work presented in this thesis.

The principle behind ToA-based schemes is straightforward. Let Δt_{2W} be the *two-way* (ToF) between interrogator and responder. The slant range r is:

$$r = \frac{1}{2} c \Delta t_{2W},$$

where $c \approx 1500$ m/s is the average sound speed along the path. c is sometimes refined iteratively by leveraging measures from on-board temperature T (°C), salinity S (‰), and depth z (m) sensors, using the Mackenzie Equation [18]:

$$c(S, T, z) \approx 1449.2 + 4.6T - 0.055T^2 + 0.00029T^3 \\ + (1.34 - 0.01T)(S - 35) + 0.016z.$$

²In underwater systems, the baseline is the known distance between two sensors or beacons.

If the source and the receiver of the signal have synchronized and stable clocks ³, Equation 2.2.2 becomes:

$$r = \frac{1}{2} c \Delta t_{1W},$$

where Δt_{1W} is the *one-way* ToF. In this case the receiver simply records the absolute transmit time (known *a priori* from its own clock) and subtracts it from the arrival time, eliminating the need for the return leg.

The most known example of ToF-based positioning scheme is Long Baseline (LBL). An LBL network comprises $N \geq 3$ (in 2D) or $N \geq 4$ (in 3D) beacons at known positions $\mathbf{p}_i^b$. The UUV (whose position is unknown) collects a slant range measurement r_i from beacon b_i ; its position is then obtained via multilateration:

$$\mathbf{p}^* = \arg \min_{\mathbf{x}} \sum_{i=1}^N (\|\mathbf{x} - \mathbf{p}_i^b\| - r_i)^2,$$

which can be solved by nonlinear least-squares solvers like Gauss-Newton or Levenberg-Marquardt.

A disadvantage of "pure" LBL system lies in the UUV having to collect at least three (in 2D) or four (in 3D) range measures to be able to localize itself. In fact, a single range defines a circle around one beacon; with two beacons, the two circles intersect in two points, so the vehicle could lie at either intersection. Only adding a third beacon, and in absence of measurement errors, uniquely fixes a vehicle's 2D position. In 3D, where each range defines a sphere around a beacon, a fourth beacon is needed to pick out the common intersection⁴.

The ranging measurement errors, however, cause the estimated position to lie in an area (or volume), rather than a point, introducing an uncertainty that is further exacerbated by the fact that the motion the UUV undergoes turns the collection time of all the necessary ranges into translational error. Hence, LBL-based techniques often couple acoustic ranging with measurement from complementary sensors or vehicle motion models that constraint the partial information provided by performing pose updates by only using a subset of the collected ranges (e.g., from $N \leq 3$ beacons) at a given time.

The localization frameworks presented in this thesis rely on an **asynchronous sequential LBL** paradigm, in which $N > 1$ beacons are present, but the vehicle's pose fixes are performed only using the partial information provided by a single slant range, as soon as an acoustic ping from a beacon is received. In order to estimate the vehicle's state (e.g., pose and velocity), the vehicle motion, the measurements from a combination of the sensors seen in Section 2.2.1 and the acoustic range information from the beacons is fused according to a *Bayesian* state estimation interpretation using different variations of probabilistic algorithms.

The next section provides the reader with the necessary toolset of probabilistic state estimation and representation at the foundations of the methods presented in this manuscript.

³Synchronization and clock stability are paramount in one-way ranging, since even a few μ s timing error translates to centimeter-level range bias.

⁴Although complementary pressure sensors can reduce the 3D problem to 2D by providing depth.

2.3 Probabilistic State Estimation and Representation

Modern autonomy and robotics instrumentation rely on the real-time interpretation of noisy, partial, asynchronous data streams and control inputs to estimate the true system state. Interpreting the problem from a probabilistic perspective allows modeling its formulation through a *Bayesian* formulation, where the posterior distribution over the state $\mathbf{x}_k$ at time step k , conditioned on all observations $\mathbf{z}_{1:k}$ and control inputs $\mathbf{u}_{1:k}$, is inferred.

The Bayesian formulation underpins both recursive filtering (e.g., the EKF [19]) and batch-based estimators (e.g., Factor Graphs [20]), that are the constituting elements of the localization frameworks proposed in this manuscript. After formalizing state estimation under the Bayesian perspective, this section delves into a brief review of the EKF and Factor Graphs estimators (Sections 2.3.2 and 2.3.3).

2.3.1 A Bayesian Perspective on State Estimation

Let $\mathbf{x}_k \in \mathbb{R}^n$ represent the state of the UUV at a discrete time index k , $\mathbf{u}_k$ the control input between $k - 1$ and k , and a sensor measurement $\mathbf{z}_k$ received at time k . Given a nonlinear *motion model*

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \quad (2.5)$$

and a nonlinear *measurement model*

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \quad (2.6)$$

where $\mathbf{w}_k$ and $\mathbf{v}_k$ are zero-mean, white Gaussian process and measurement noises with covariances $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively. Estimating the state at time k equates to estimating the posterior:

$$p(\mathbf{x}_{0:K} \mid \mathbf{z}_{1:K}),$$

where $\mathbf{x}_{0:K} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_K\}$ and $\mathbf{z}_{1:K} = \{\mathbf{z}_1, \dots, \mathbf{z}_K\}$. The posterior computation can be carried out according to two main paradigms.

Recursive Filtering (EKF) Computing the distributions

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \quad \text{for } k = 0, 1, \dots, K,$$

is carried out in an *online* fashion. The filter is decomposed into a time update (*Prediction*) and a measurement update (*Update*) as:

$$\text{Prediction:} \quad p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1},$$

$$\text{Update:} \quad p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \propto p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}).$$

Batch Estimation (Factor Graphs) The entire trajectory $\mathbf{x}_{0:K}$ (or a subset) is estimated by maximizing the joint posterior

$$p(\mathbf{x}_{0:K} \mid \mathbf{z}_{1:K}) \propto p(\mathbf{x}_0) \prod_{k=1}^K p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \prod_{k=1}^K p(\mathbf{z}_k \mid \mathbf{x}_k).$$

whose negative logarithm allows the problem to be solved as a nonlinear least-squares problem of the form:

$$-\ln p(\mathbf{x}_{0:K} \mid \mathbf{z}_{1:K}) = \underbrace{\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\|_{\Sigma_0}^2}_{\text{prior}} + \sum_{k=1}^K \underbrace{\|f(\mathbf{x}_{k-1}, \mathbf{u}_k) - \mathbf{x}_k\|_{\mathbf{Q}_k}^2}_{\text{motion factor}} + \sum_{k=1}^K \underbrace{\|h(\mathbf{x}_k) - \mathbf{z}_k\|_{\mathbf{R}_k}^2}_{\text{measurement factor}}.$$

Filtering only maintains a single Gaussian approximation $\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k, \mathbf{P}_k)$ of the state $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ at each time k , and updates $\hat{\mathbf{x}}_{k-1} \rightarrow \hat{\mathbf{x}}_k$ via linearization and covariance propagation. Its computational complexity ($O(n^3)$ per step in the state dimension n) allows it to run *online*.

Batch-based state estimation, instead, incrementally approximates the full trajectory $\mathbf{x}_0, \dots, \mathbf{x}_K$. By posing the problem as a nonlinear sparse least squares problem, modern solvers exploit the sparsity structure of the factor graph to obtain near real-time updates within a sliding window. However, the optimization problem that is solved or updated grows larger at each step, therefore it requires careful reasoning about the sliding window size, when running on constrained systems.

The Section proceeds to take a more detailed look at the EKFs and Factor Graphs estimation frameworks.

2.3.2 Extended Kalman Filters

The EKF is the extension of the Kalman Filter (KF) under the scenario where the dynamics and measurement relationships are nonlinear. It recursively approximates the posterior $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ as Gaussian as:

$$\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_k, \mathbf{P}_k),$$

where $\hat{\mathbf{x}}_k$ is the state mean estimate, and $\mathbf{P}_k$ is the associated covariance. Because f and h in the *process* (Equation 2.5) and *measurement* (Equation 2.6) models, respectively, are nonlinear, the EKF linearizes the current state estimate $\hat{\mathbf{x}}_{k-1}$ (in the *prediction*) and $\hat{\mathbf{x}}_k^-$ (in the *update*) by their first-order Taylor expansions through the Jacobians:

$$\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}, \mathbf{u}=\mathbf{u}_k}, \quad \mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-}.$$

In the predict time update, the State-Mean propagation is computed as:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k),$$

and the covariance is propagated through:

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^\top + \mathbf{G}_k \mathbf{Q}_{k-1} \mathbf{G}_k^\top,$$

where $\mathbf{G}_k$ is the Jacobian of f with respect to the process noise $\mathbf{w}_{k-1}$. The *predict* step yields the *a priori* estimate $(\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-)$.

When a measurement $\mathbf{z}_k$ is obtained, the EKF computes the *residuals* as:

$$\mathbf{y}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_k^-).$$

For example, if $h(\cdot)$ returns a range measurement to a known acoustic beacon at position $\mathbf{p}_{\text{beacon}}$, then

$$h(\hat{\mathbf{x}}_k^-) = \|\mathbf{p}_k^- - \mathbf{p}_{\text{beacon}}\|,$$

and $\mathbf{y}_k$ is the difference between the actual measured range and the predicted range. The *Innovation Covariance* $\mathbf{S}$ and *Kalman Gain* $\mathbf{K}$ are then computed as:

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^\top + \mathbf{R}_k, \quad \mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^\top \mathbf{S}_k^{-1},$$

where, $\mathbf{R}_k$ is the measurement noise covariance matrix. Finally, the State-Mean Update of the measurement update step is:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{y}_k.$$

and the covariance update is:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^\top + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^\top,$$

computed using the Joseph form to maintain numerical symmetry in $\mathbf{P}_k$.

2.3.3 Factor Graphs

Factor Graphs represent an alternative method to recursive filtering for state estimation in which the entire trajectory (or a sliding window) of states is estimated jointly by posing the problem as a nonlinear least-squares optimization. They represent the factorization of a joint probability distribution over states and measurements through a graph-based graphical model.

Let $k = 0, 1, 2, \dots, K$ be a sequence of discrete timesteps, and:

- $\{\mathbf{x}_k\}_{k=0}^K$ the set of *state variables*, e.g., position $\mathbf{p}_k$, orientation $\mathbf{q}_k$, velocity $\mathbf{v}_k$, and any biases.
- $\{\mathbf{z}_k^{(m)}\}$ the set of *measurement variables*, indexed by sensor type m .
- $\Phi = \{\phi_i\}$ the set of *factors*, each factor ϕ_i representing a probabilistic constraint (likelihood or prior) that connects one or more state variables.

Then, a Factor Graph is a *bipartite graph* $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$, where:

- $\mathcal{V} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_K\}$ is the set of variable nodes.
- $\mathcal{F} = \{\phi_i\}$ is the set of factor nodes.
- $\mathcal{E} \subset \mathcal{V} \times \mathcal{F}$ is the set of edges, with an edge $(\mathbf{x}_k, \phi_i)$ *iff* factor ϕ_i involves variable $\mathbf{x}_k$.

Each factor $\phi_i(\mathcal{X}_i)$ is a nonnegative function of the subset of variables $\mathcal{X}_i \subseteq \mathcal{V}$. In a maximum-a-posteriori (MAP) formulation, the joint probability $p(\{\mathbf{x}_k\} \mid \{\mathbf{z}\})$ factorizes as:

$$p(\{\mathbf{x}_k\} \mid \{\mathbf{z}\}) \propto \prod_i \phi_i(\mathcal{X}_i).$$

Equivalently, taking the negative log yields a sum of residuals:

$$-\ln p(\{\mathbf{x}_k\} \mid \{\mathbf{z}\}) = \sum_i \underbrace{\|\mathbf{r}_i(\mathcal{X}_i)\|_{\Sigma_i}^2}_{\text{error term from } \phi_i} + \text{const.},$$

where $\mathbf{r}_i(\mathcal{X}_i)$ is the residual associated with factor ϕ_i , and Σ_i is its covariance. Residuals are collected into a vector $\mathbf{r}(\mathbf{x})$, where $\mathbf{x} = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_K]^\top$. The MAP estimate is obtained by solving:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \sum_i \|\mathbf{r}_i(\mathcal{X}_i)\|_{\Sigma_i}^2.$$

This is typically a large nonlinear least-squares problem, which is generally solved through nonlinear optimization techniques such as Gauss–Newton, Levenberg–Marquardt, or Incremental Smoothing and Mapping (iSAM) [21] to compute a MAP estimate $\{\mathbf{x}_k^*\}$.

iSAM2

Batch factor-graph optimization is highly accurate, but comes at the runtime cost of rebuilding and refactorizing the full linear system at every step. iSAM2 [22], instead, allows maintaining (close to)-batch accuracy while updating the solution in logarithmic time by exploiting the structure of the factor graph.

The algorithm performs the inference in two stages:

1. **Linearisation.** Each residual is first-order approximated around the current linearization point $\bar{\mathbf{x}}$ as

$$\mathbf{r}_i(\mathcal{X}_i) \approx \mathbf{A}_i \delta \mathbf{x}_{\mathcal{X}_i} - \mathbf{b}_i, \quad (2.7)$$

where $\delta \mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$.

Collecting all factors yields the normal equations

$$\mathbf{A}^\top \Sigma^{-1} \mathbf{A} \delta \mathbf{x} = \mathbf{A}^\top \Sigma^{-1} \mathbf{b}. \quad (2.8)$$

2. **Variable elimination.** Sparse Cholesky factorization (variable elimination) converts the linear factor graph into a *Bayes tree*.⁵ Because cliques are conditionally independent given their separators, only those affected by new factors need to be re-linearized and re-factorized.

Incremental update steps. At time step $k+1$, new motion/measurement factors Φ_{new} are added. Introducing Φ_{new} marks as "contaminated" every clique that involves a newly touched variable; their union forms the "affected" subtree $\mathcal{T}_{\text{aff}}$. Only variables in $\mathcal{T}_{\text{aff}}$ are re-linearised. Re-eliminating the updated linear factors regenerates Gaussian conditionals for $\mathcal{T}_{\text{aff}}$, which is then put back onto the unchanged remainder of the tree. A root-to-leaf back-substitution pass updates the optimal increments $\delta \mathbf{x}$ only along changed paths, yielding $O(|\mathcal{T}_{\text{aff}}| \log |\mathcal{T}|)$ time per update.

⁵Bayes trees are junction-trees whose cliques store conditional Gaussians. They encode factored probability densities.

Managing non-linearity. Since residuals are locally linearized, iSAM2 monitors the increment of each variable. If $\|\delta \mathbf{x}_i\|$ exceeds a threshold, that variable is marked for re-linearization in the next cycle, guaranteeing eventual consistency with a fully re-linearized batch solution.

Fixed-lag smoothing and marginalization. To bound memory usage, iSAM2 supports *fixed-lag* smoothing, in which only states older than a horizon τ are marginalized. Marginalization eliminates the outdated variables and introduces a prior on their Markov blanket ⁶, preserving information while keeping the Bayes tree shallow.

Therefore, iSAM2 retains the probabilistic fashion of batch estimation but can achieve online performance sometimes close to recursive filters.

The following Chapter 3 introduces methods based on EKF formulations, whereas Factor Graphs and iSAM2 are employed in Appendix A.

⁶I.e., the minimal set of variables that make the variable conditionally independent of all the variables in the tree

Chapter 3

Acoustic Localization of Autonomous Underwater Vehicles

Increasing commercial and strategic interests in the sustained and sustainable exploitation, preservation and regeneration of the marine environment has recently propelled the whole field of underwater communication and networking to unparalleled advances. The driving sectors of the *Blue Economy*, including tourism, climate change, renewable energy, aquaculture, waste management, security and maritime transport, now more than ever need the support of networking solutions at par with those available on earth, to ease swift data reporting, real-time monitoring, and other management operations.

Many underwater applications require use of untethered devices that are capable of communicating wirelessly, thus offsetting the cost of deploying cables underwater that would be prohibitive for many applications, and that enable asset deployment in large geographical areas. Particularly, mobile autonomous devices such as AUVs are effective for decreasing costs, e.g., of experimental, surveillance, monitoring and exploration campaigns, for reducing human intervention in hazardous areas, and for extending network coverage.

Localizing AUVs clearly increases their usefulness for many applications. However, this is particularly challenging underwater, for the impossibility of relying on radio communication and especially on positioning system like the Global Positioning System (GPS).

To tackle this challenge, a variety of technologies and techniques (see Section 2.2) have been introduced, ranging from passive ones that are relatively inexpensive but inaccurate like Dead Reckoning and Inertial Navigation, to increasingly accurate ones, relying actively on communications, more expensive equipment, sophisticated on board sensors and requiring longer deployment and calibration times, such as acoustic or optical-based localization, and geophysical navigation [23, 24].

Underwater acoustic systems (see Section 2.2.2) appear to be particularly suited to localize underwater assets, including mobile ones [25]. These systems allow estimating the vehicle state (e.g., position, orientation, velocity) by means of acoustic distance and direction measurements. Some of these systems require acoustic transponders

to be deployed on a ship or similar vessel (e.g., Short Baseline (SBL) [26, 27], Ultra-Short Baseline (USBL) [28–31] or their combinations [32, 33]). However, the work presented in this Chapter is based on LBL to determine asset locations relative to a framework of baseline stations, which must be deployed prior to operations, and whose positions need to be known [25] (Ch. 4).

In fact, LBL-based systems do not need to rely on ship support, making them suitable for localizing mobile vehicles, especially those used in campaigns at sea, with small form factors and room for little payloads and very basic sensors (e.g., Inertial Measuring Units (IMUs) or depth sensors).

Although systems of this kind are usually more precise, a trade-off is to be made between accuracy and both cost (of devices and their deployment) and time (for deployment and calibration). Accurate LBL systems offer a clear example: setting the beacons up, and providing them with accurate ways to keep up-to-date coordinates can be costly, especially when their use is requested only for a limited period of time. This is why, when cost is an issue, and whenever a localization system is needed swiftly or temporarily, new methodologies are sought after that provide acceptable accuracy “on the fly,” possibly using off-the-shelf components.

One of such methodologies uses optimized statistical processing of sensed data to cope with the nonlinear dynamics of the environment. Kalman filtering and its variants play an essential role in these scenarios by aggregating the measures and the bias associated to the system components into a comprehensive estimate of the system state.

However, the accuracy of the estimates obtained by using Kalman filters is highly dependent on the a priori knowledge of the noise statistics of the model. The use of incorrect statistics can lead to substantial estimation errors or even to filter divergence, especially if the noise is time-varying, as it is often the case underwater. Adaptive noise estimation techniques allow such issues to be obviated by estimating the error statistics of the vehicle state or of its sensors while moving. It follows that systems that are able to adapt to the time-varying noise conditions of the environment or of the instrumentation are less dependent on expensive sensor suites and costly equipment (that are more accurate and noise-resilient by design) if they favor, instead, active noise compensation strategies.

The work presented in this Chapter demonstrates these concepts in experimental settings at sea. Starting from a minimal-setup non adaptive localization framework with static noise settings (derived from the published work in [34]), this thesis highlights, although proper localization accuracy in some scenarios is possible, the drawbacks of not being able to correct for the time-varying noise statistics in marine environment.

Anchored to the same concept of enabling localization on minimal-setup vehicles and architectures, the thesis then addresses such drawbacks by presenting a solution (derived from the published work in [35]) that is able to improve the vehicle’s position estimates accuracy through various noise estimation and auto-calibration steps.

In detail, the rest of this Chapter is structured as follows:

- Section 3.1 describes the experimental settings, the architecture, equipment

and environmental parameters common to all the methods presented in this Chapter;

- Section 3.2 presents a range-only EKF-based localization method for small form-factor AUVs with minimal equipment and setup/calibration times. It concerns a solution for time and cost efficient at-sea setups, in which no sensor suite on board the AUV is employed and the beacons' positions are rapidly surveyed with a general handheld GPS as soon as they are deployed, without further calibration operations. The analysis of the evaluation metrics of this system will serve as a baseline for the subsequent presented work;
- In Section 3.3 the same setup is augmented with an initial beacon position auto-calibration that allows the AUV to further refine its position estimate when ranges are received, and with the measurements from the on board IMU. Moreover, this method implements adaptation techniques that allow the filter to perform online adjustments to the measurement (range) noise covariance and to calibrate the initial bias on the heading measures derived from the on board IMU.

3.1 Experimental Settings and Scenario

The field experiments in which the following presented methods are evaluated have been carried out around the *Fortino* inlet, a bay on the eastern side of the island of Ponza in the context of Project "*TEAM: Tecnologie Emergenti per l'Archeologia Marina*" [36] with the goal of acquiring images and *sonar*-based reconstructions of a shipwreck (see Figure 3.1). The bathymetry (≤ 15 m) and a low sea state (1 on the

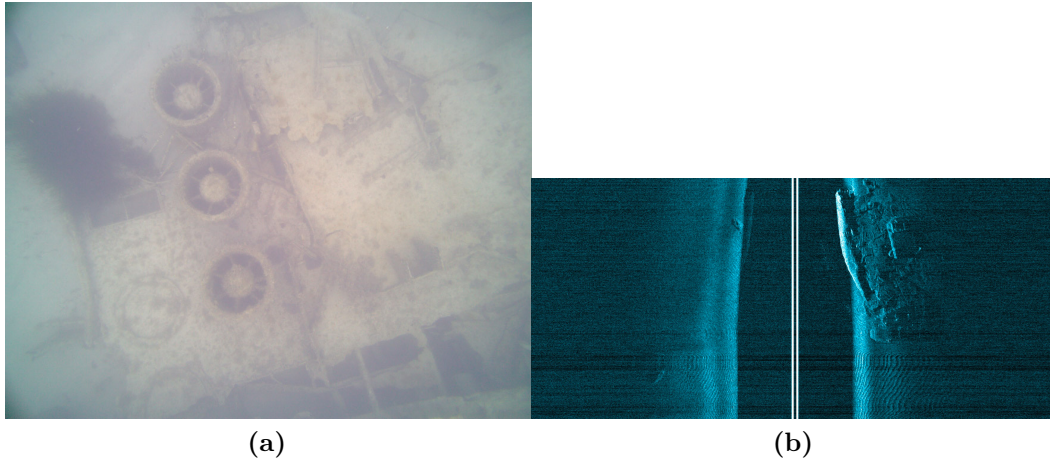


Figure 3.1. (a) Camera capture of the shipwreck. (b) Wreck mapping (right side of the scan) through a Side Scan Sonar.

Douglas scale) allowed for controlled navigation trials.

Four acoustic transponders (beacons) were suspended 1.5 m below surface buoys and anchored to the seabed. Their deployment geometry is shown in Figure 3.2. The

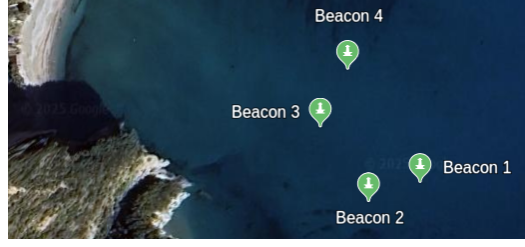


Figure 3.2. Aerial view of the beacon deployment.

acoustic beacons are WNodes commercialized by WSense Srl with the specifications shown in Table 3.1.

Frequencies	24–32 kHz
Data Rate	640 bps
Source Level	168 dB re 1 μ Pa @ 1 m
Operating Depth	Up to 300 m
Battery	147 Wh
TX Power Consumption	0.5 W
Idle Power Consumption	66.5 mW
Power Consumption in Sleep Mode	665 μ W

Table 3.1. Technical specifications of the WNode’s acoustic modem.

The geodetic coordinates of each deployed beacon have been surveyed with a low-precision (± 4 m horizontal positioning error) handheld GPS (although not all the presented methods use this information).

The AUV employed in the sea trials is the **Zeno** (Zeno Environmental Nautical Operator) by MDM Team [37,38] (in Figure 3.3). The vehicle (of approximate length



Figure 3.3. The Zeno AUV.

1.2 m) carried an Intel NUC running ROS Melodic [39], a rear thruster, rudder and elevator. Its navigation sensors comprised ¹:

- Ublox NEO-8T GPS (surface only, 1Hz update frequency, ± 2.5 m horizontal positioning error);

¹The methods presented in this Chapter only use different subsets of the available sensor suite.

- pressure transducer (0.01 m resolution, 10 Hz update frequency);
- 9-Degrees-of-Freedom (DoF) IMU (10 Hz update frequency);
- A DVL (1 Hz update frequency);
- A WSense acoustic modem running the SUNSET Software Defined Communication Stack (SDCS) [40] enabling the network protocol stack for message exchange and localization.

Figure 3.4 shows the Zeno AUV during the experiments, integrated with the WSense acoustic modem. During the mission, the AUV navigated at a mean speed of 0.3 m/s



Figure 3.4. (a) WSense acoustic modem integrated in the hull of the Zeno AUV. (b) The Zeno AUV during navigation in the waters of Ponza.

and never below a depth 0.3 m (in order to receive GPS fixes for position and velocity ground-truth) for about 50 minutes, tracing the trajectory shown in Figure 3.5 and polling, at a frequency of 0.5 Hz, each beacon to obtain Two-Way range information (see Section 2.2.2). The timeout on the polling messages from the AUV is 1 s , i.e. if after 1 s no beacon has responded to the acoustic poll by the AUV, the packet is discarded.

All the methods presented in the rest of this Chapter perform this ranging procedure as soon as they are deployed and until the end of the mission. Figure 3.6 shows the acoustic ranges computed by the AUV during the entire mission, from each of the deployed beacons. As can be noted, the AUV is never out of the communication range of any beacon, and the maximum distance from a beacon reached during the mission is about 88 m.

Table 3.2 summarises the key experimental parameters.

The following Section introduces the first of the two localization methods proposed in this Chapter.

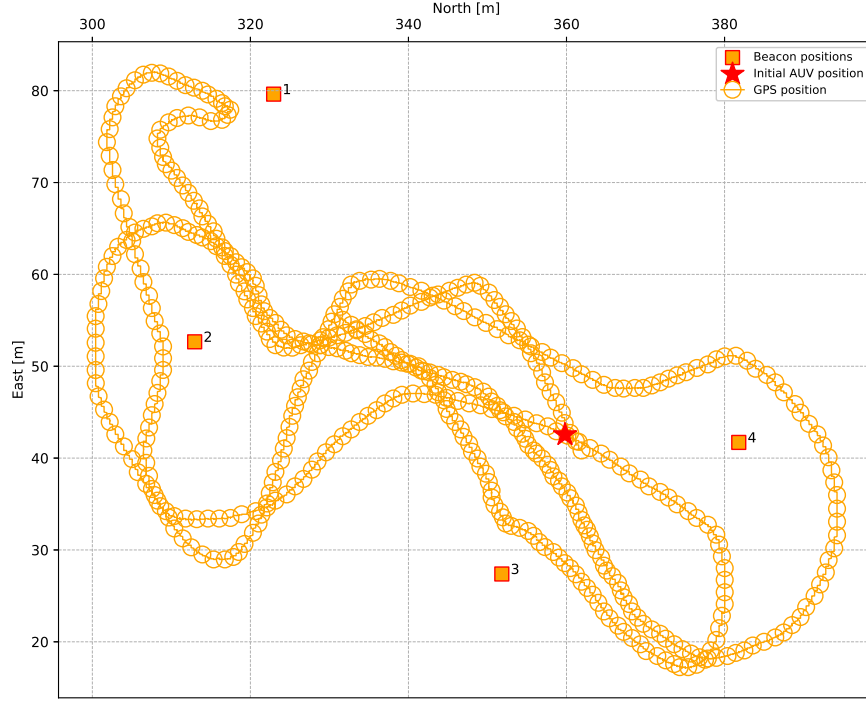


Figure 3.5. Mission trajectory measured by the onboard GPS and beacons positions.

3.2 Experimental At-Sea Evaluation of a Baseline EKF-Based Localization Approach

This Section lays the technical groundwork for a light-weight, beacon-based localization pipeline that fits the constraints of small AUVs while still delivering meter-level accuracy in the described scenario. The presented method relies on a simple estimator: a time-invariant EKF that fuses (i) depth from the pressure sensor, (ii) slant ranges from an acoustic LBL network, and (iii) a constant-velocity motion prior perturbed by Brownian acceleration. This minimal setup avoids the payload and calibration overhead typical of more complex sensor suites, yet, as it will be demonstrated, it is still able to achieve sub-3 m Root Mean Square Error (RMSE) on a real-world sea trial.

The Section is organized as follows.

- Section 3.2.1 formalizes the discrete-time kinematics and derives the EKF prediction–update recursion together with the linearized Jacobians required for the real-time implementation.
- Section 3.2.2 benchmarks the filter on a coastal survey and quantifies its absolute localization error against ground-truth GPS fixes along other filter accuracy and stability metrics.

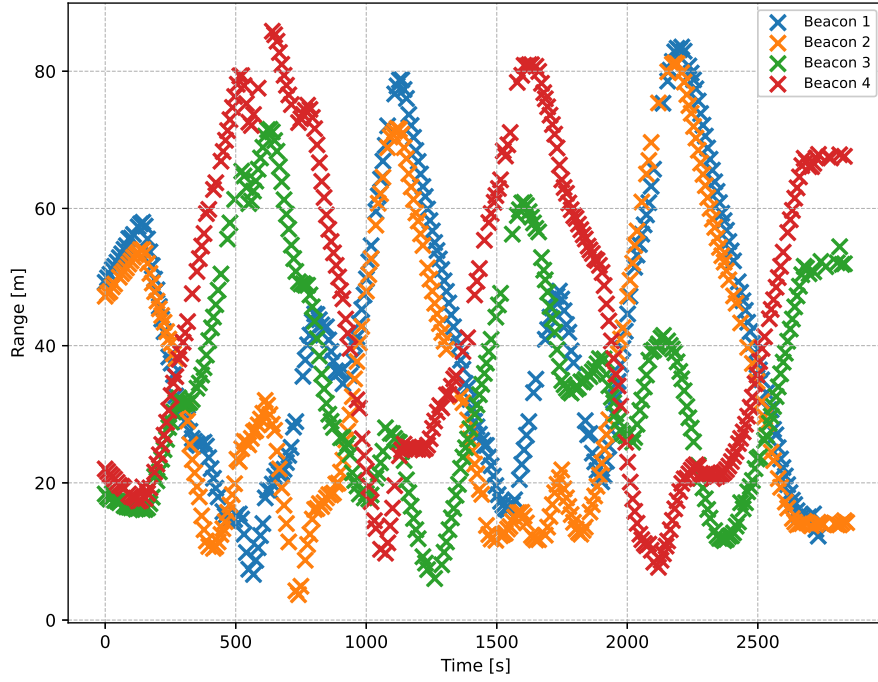


Figure 3.6. Acoustic ranges measured by the AUV's modem during the mission.

- Section 3.2.3 interprets the evaluation done in the previous Section to uncover the main limitations of a fixed-tuned filter in a non-stationary and error-prone environment.
- Section 3.2.4 positions the presented method in the state-of-the-art of approaches of similar class.

The evidence gathered here will serve a dual purpose:

1. establishing a reference against which more sophisticated adaptive estimators will be judged in the remainder of the thesis;
2. isolating the two factors (time-varying measurement noise and beacon position uncertainty) that most critically hinder performance in harsher conditions.

Addressing these factors motivates the adaptive extensions introduced in Chapter 3.3.

3.2.1 System Model

Motion model: constant-velocity with Brownian acceleration. Let the AUV's planar position-velocity state be $\mathbf{x}_k = [\mathbf{p}_k^\top \mathbf{v}_k^\top]^\top \in \mathbb{R}^4$, with $\mathbf{p}_k = (p_{x,k}, p_{y,k})^\top$ and $\mathbf{v}_k = (v_{x,k}, v_{y,k})^\top$. Assuming that between two control-cycle instants ($t_{k-1} \rightarrow$

Parameter	Value / Notes
Site	Fortino Bay inlet, Ponza (Italy)
Bathymetry	10–15 m, sea-state 1
Test area	$\sim 80 \times 100$ m rectangle
Beacon depth	1.5 m below buoys
Beacon positions	Handheld GPS (± 4 m)
AUV sensors	GPS, pressure, 9-DoF IMU, acoustic modem, DVL
Polling schedule	Round-robin, 1 s timeout per beacon
Filtering rates	Propagation 10 Hz; updates on every range
Mission duration	50 min

Table 3.2. Summary of Ponza field campaign settings.

$t_k = t_{k-1} + T$) the vehicle keeps a constant mean velocity, while any manoeuvre is absorbed as a zero-mean white Gaussian acceleration, the continuous kinematics $\dot{\mathbf{p}}(t) = \mathbf{v}(t)$, $\dot{\mathbf{v}}(t) = \mathbf{a}(t)$ integrate to the discrete stochastic recursion:

$$\boxed{\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k}, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}),$$

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = a \begin{bmatrix} \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 \\ 0 & \frac{T^4}{4} & 0 & \frac{T^3}{2} \\ \frac{T^3}{2} & 0 & T^2 & 0 \\ 0 & \frac{T^3}{2} & 0 & T^2 \end{bmatrix}, \quad a \in \mathbb{R}^+.$$

where $\mathbf{Q}$ arises from integrating white acceleration of spectral density a over T . The upper-left $T^4/4$ entries measure how double-integration of noise spreads position, the lower-right T^2 block is the velocity variance, and the symmetric $T^3/2$ terms tell that a single noisy burst of thrust perturbs both the velocity and position at the end of the interval.

Slant-range observation model with composite noise. When beacon i replies, the AUV computes a slant range $d_{i,k}$ estimate (see Section 2.2.2). The on-board depth sensor supplies the vehicle depth z_k . Given the (GPS-surveyed) beacon coordinates $\mathbf{p}_i = (p_{i,x}, p_{i,y}, p_{i,z})^\top$, combining the two gives the nonlinear measurement equation:

$$d_{i,k} = h_i(\mathbf{x}_k, \mathbf{r}) + r_{r,k}$$

$$h_i(\mathbf{x}_k, \mathbf{r}) = \sqrt{\|\mathbf{p}_k - \mathbf{p}_i + \mathbf{r}_b\|^2 + (z_k - p_{i,z} + r_{b,z})^2},$$

where the zero-mean composite error $\mathbf{r} = [\mathbf{r}_b^\top r_r]^\top \in \mathbb{R}^4$ groups the beacon-position uncertainty $\mathbf{r}_b = (r_{b,x}, r_{b,y}, r_{b,z})^\top$ and the range-sensor noise r_r . The measurement noise covariance matrix $\mathbf{R}$ such that $\mathbf{r} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ is:

$$\mathbf{R} = \text{diag}(\sigma_b^2 \mathbf{I}_3, \sigma_r^2),$$

EKF recursion. At a frequency of 10 Hz (each time a measure is received from the depth sensor), the vehicle performs the *Prediction* step of the EKF as:

$$\begin{aligned} \text{State Time Update: } \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1} \\ \text{Covariance Propagation: } \mathbf{P}_{k|k-1} &= \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^\top + \mathbf{Q} \end{aligned}$$

Each time a slant range $d_{i,k}$ is computed (~ 0.5 Hz) the *Update* step of the EKF is triggered as:

$$\begin{aligned} \text{Residual Computation: } \mathbf{y}_k &= \mathbf{d}_k - h(\hat{\mathbf{x}}_{k|k-1}) \\ \text{Residual-Space Covariance: } \mathbf{S}_k &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^\top + \mathbf{M}_k\mathbf{R}\mathbf{M}_k^\top \\ \text{Kalman Gain: } \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^\top\mathbf{S}_k^{-1}, \\ \text{State Update: } \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\mathbf{y}_k \\ \text{Covariance Update: } \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}, \end{aligned} \tag{3.1}$$

where $\mathbf{H}_k \in \mathbb{R}^{n \times 4}$ and $\mathbf{M}_k \in \mathbb{R}^{n \times 4}$ in Equation (3.1) are the row-stacked (for n ranges, in this section $n = 1$) linearization Jacobians:

$$\mathbf{H}_{i,k} = \frac{\partial h_i}{\partial \mathbf{x}} = \begin{bmatrix} \frac{s_x}{r_{i,k}} & \frac{s_y}{r_{i,k}} & 0 & 0 \end{bmatrix}, \quad \mathbf{M}_{i,k} = \frac{\partial h_i}{\partial \mathbf{r}} = \begin{bmatrix} \frac{s_x}{r_{i,k}} & \frac{s_y}{r_{i,k}} & \frac{s_z}{r_{i,k}} & 1 \end{bmatrix},$$

with $\mathbf{s}_{i,k} = (s_x, s_y, s_z)^\top = \mathbf{p}_k - \mathbf{p}_i + \mathbf{r}_b$ and $r_{i,k} = h_i(\mathbf{x}_k, \mathbf{r})$.

The next Section evaluates this model in the scenario described in Section 3.1.

3.2.2 Experimental Evaluation

The presented localization pipeline is hereby evaluated. For all the duration of the mission, at each time k , the AUV collected each GPS measurement (only used for Ground-Truth) and the state estimates $\mathbf{p}_k$ and $\mathbf{v}_k$. The localization error is evaluated by computing the RMSE between the GPS-acquired positions and the EKF-estimated positions as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad \begin{aligned} \text{where } y_i &= \text{ground-truth value,} \\ \hat{y}_i &= \text{estimated value,} \\ N &= \text{number of samples.} \end{aligned}$$

The noise parameters embedded in the EKF formulation are shown in Table 3.3. Figure 3.7 shows the 2D estimated AUV trajectory versus that of the one estimated by the GPS and the computed Euclidean localization error in time, along with the RMSE.

Noise source	Sym.	Distrib.	Std./Param.	Value in exp.
Process acceleration	$\mathbf{w}$	$\mathcal{N}(0, \mathbf{Q})$	$\sqrt{a}$ (m s ⁻²)	$T = 0.1$ s, $a = 0.1$
Beacon placement	$\mathbf{r}_b$	$\mathcal{N}(0, \mathbf{R})$	$\sigma_b^2 = 1.2$ m ²	same for x, y
Ranging error	r_r	$\mathcal{N}(0, \mathbf{R})$	$\sigma_r^2 = 1.2$ m ²	independent of $\mathbf{q}_b$
Depth-sensor error	—	—	0 (treated error-free)	sensor accuracy $\ll 0.05$ m

Table 3.3. Noise statistics used in the present EKF formulation.

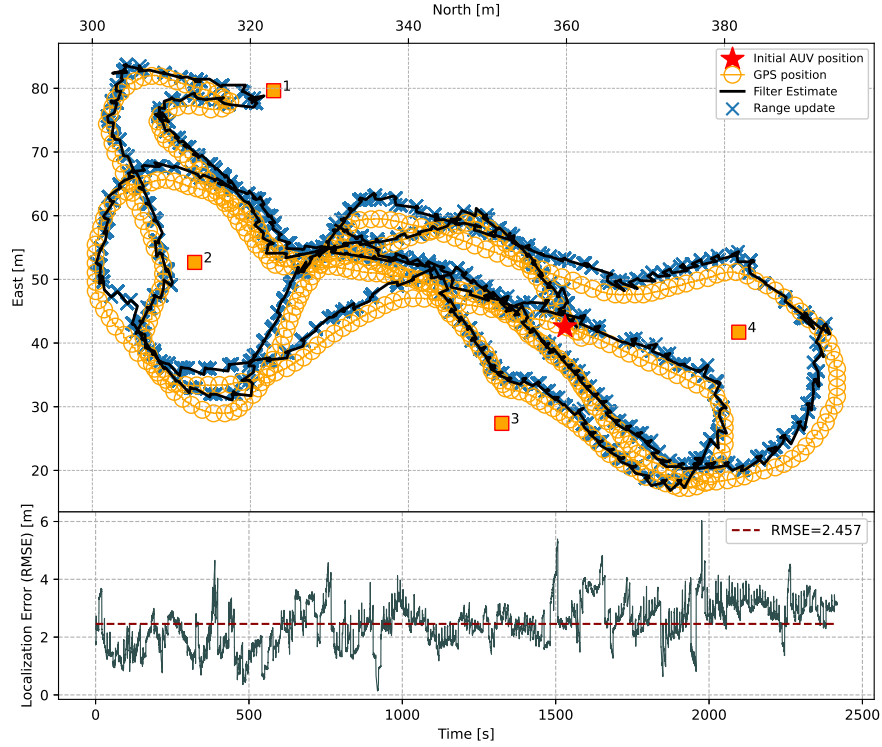


Figure 3.7. 2D Trajectory estimated by the proposed method.

As can be seen, the RMSE achieved by this simple method is ~ 2.46 m.

This EKF formulation is also able to estimate the unobserved velocity because (i) the kinematic matrix $\mathbf{F}$ mixes velocity into position, (ii) the pair $(\mathbf{F}, \mathbf{H}_k)$ is observable, and (iii) process noise keeps the velocity–position coupling active in the covariance. Thus, every range correction in the Update step also allows refining velocity as well as position. More formally, the constant-velocity/Brownian-acceleration assumption gives the discrete linear dynamics:

$$\mathbf{x}_k = \underbrace{\begin{bmatrix} \mathbf{I}_2 & T\mathbf{I}_2 \\ \mathbf{0}_2 & \mathbf{I}_2 \end{bmatrix}}_{\mathbf{F}} \mathbf{x}_{k-1} + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}),$$

where the upper-right block is $T\mathbf{I}_2$, the previous velocity $\mathbf{v}_{k-1}$ is "copied" into the next position $\mathbf{p}_k$. When a single acoustic beacon returns one slant range $d_k = h(\mathbf{x}_k, \mathbf{r}) + r_{r,k}$, linearizing around the predicted state yields the Jacobian

$$\mathbf{H}_k = \frac{\partial h}{\partial \mathbf{x}} = \begin{bmatrix} \frac{s_x}{r_k} & \frac{s_y}{r_k} & 0 & 0 \end{bmatrix}, \quad (s_x, s_y) = \mathbf{p}_k - \mathbf{p}_i + \mathbf{r}_b.$$

Stacking ² the Jacobians produced by the *same* beacon over consecutive instants (four in this example):

$$\mathcal{O} = \begin{bmatrix} \mathbf{H}_k \\ \mathbf{H}_k \mathbf{F} \\ \mathbf{H}_k \mathbf{F}^2 \\ \mathbf{H}_k \mathbf{F}^3 \end{bmatrix} = \begin{bmatrix} h_x & h_y & 0 & 0 \\ h_x & h_y & Th_x & Th_y \\ h_x & h_y & 2Th_x & 2Th_y \\ h_x & h_y & 3Th_x & 3Th_y \end{bmatrix},$$

with $h_x = s_x/r_k$ and $h_y = s_y/r_k$. Because $T \neq 0$ and not both h_x, h_y are zero, the four rows are linearly independent; therefore $\text{rank}(\mathcal{O}) = 4$ and the full state $[\mathbf{p}^\top \mathbf{v}^\top]^\top$ is (locally) observable using one beacon alone. Being the process-noise covariance $\mathbf{Q} \succ \mathbf{0}$, the predicted covariance keeps the cross term $\frac{T^3}{2}\mathbf{I}_2$ between position and velocity.

Consequently the Kalman gain has a non-zero velocity block:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1} = \begin{bmatrix} * \\ \frac{T^3}{2} h/r_k \end{bmatrix},$$

so every range innovation Δd produces a velocity correction $\Delta \mathbf{v}_k = \mathbf{K}_k^{(v)} \Delta d$. Figure 3.8 shows the North and East velocity components estimated by the EKF against the (noisy) velocity components measured by the on board DVL.

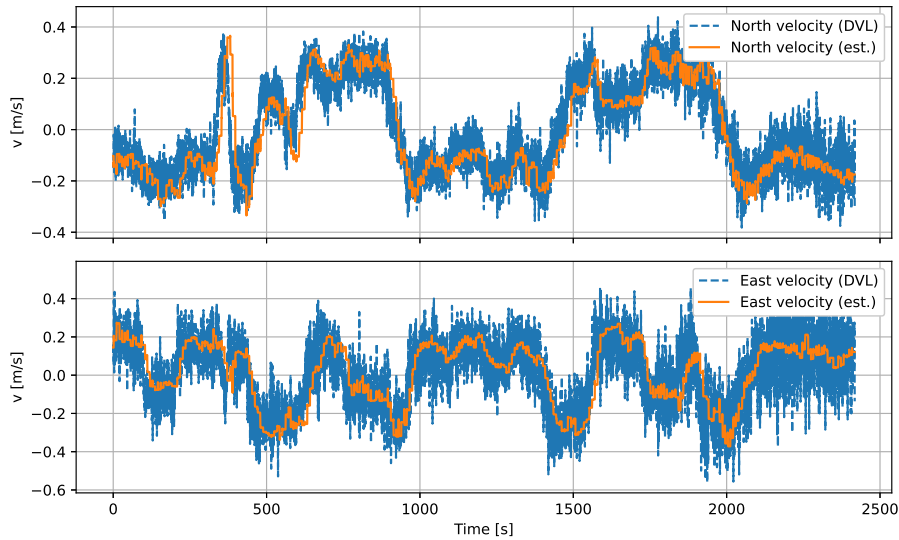


Figure 3.8. 2D velocity estimated by the proposed method.

²This is shown as stacking for the sake of an intuitive explanation; in reality, the EKF recursively accumulates this information using a single range at a time.

Figure 3.9 aggregates the visual representations of several statistical analysis carried out on the accuracy and consistency of the presented EKF, and will be discussed in the following.

Figure 3.9c shows the empirical Cumulative Distribution Function (CDF) of the 3D Euclidean position error $\varepsilon_k = \|\hat{\mathbf{x}}_{k|k} - \mathbf{x}_k^{\text{truth}}\|$. The 95th percentile occurs at 3.49 m, hence the majority of localization error sit fairly below that, being the RMSE over the entire run 2.46 m, as previously shown.

Consistency is first checked in the innovation space via the Normalised Innovation Squared (NIS) $\text{NIS}_k = \mathbf{y}_k^\top \mathbf{S}_k^{-1} \mathbf{y}_k$, where $\mathbf{y}_k$ and $\mathbf{S}_k$ are the scalar range residual and its predicted covariance, respectively. For a correctly tuned filter NIS_k should follow a χ_1^2 law. Figure 3.9d reports that 94.5 % of the samples fall inside the two-sided 95 % χ_1^2 envelope. Hence the assumed measurement noise R and process noise Q are jointly credible.

Figure 3.9a overlays a Rayleigh Probability Density Function (PDF) on the histogram of ε_k . Up to ≈ 2.5 m the fitted scale parameter $\hat{\sigma} = 1.7$ m closely matches the data, confirming that the bulk error behaves like the magnitude of a zero-mean Gaussian vector. A Quantile-Quantile (Q-Q) plot (Figure 3.9b) reveals a heavier right tail: the largest 5 % of errors are ~ 0.5 –1 m larger than a pure Rayleigh model would predict.

Figure 3.9e tracks a 100-sample rolling RMSE together with the corresponding rolling 95 % error. Both curves oscillate between 1.4 m and 5 m. Critically, the gap between the two curves remains narrow, indicating that the error distribution retains the same shape and only its scale changes. Finally, the State Covariance Matrix $\mathbf{P}$ stayed positive definite for the whole duration of the experiment.

In the next Section, the evidence resulting from the performed analysis are discussed and elaborated to highlight the principal aspects that can be optimized to inform a better state estimation strategy.

3.2.3 Results Discussion

The experimental results of Section 3.2.2 show that a classical, time-invariant EKF fed by depth, slant-range and a constant-velocity motion prior can already deliver a proper localization accuracy (RMSE $\simeq 2.46$ m) for many applications. While this performance is adequate for many short-term or low-dynamics surveys, three practical factors limit its robustness and scalability to harsher field conditions.

The diagnostic evidence presented in the previous Section motivates two main complementary upgrades to the presented EKF: making the measurement covariance $\mathbf{R}_k$ adaptive, and refining the belief about beacon positions.

In fact, although the NIS test passes globally (94.5 % in-bounds, Figure. 3.9d), small clusters of spikes still occur, suggesting that adapting $\mathbf{R}_k$ might capture events that make the normalized innovation exceed a certain value.

Also, the Q-Q curve in Figure 3.9b bends above the Rayleigh line, indicating larger residuals than a stationary Gaussian model predicts. An epoch-wise variance estimator might be able to down-weight such episodes in real time.

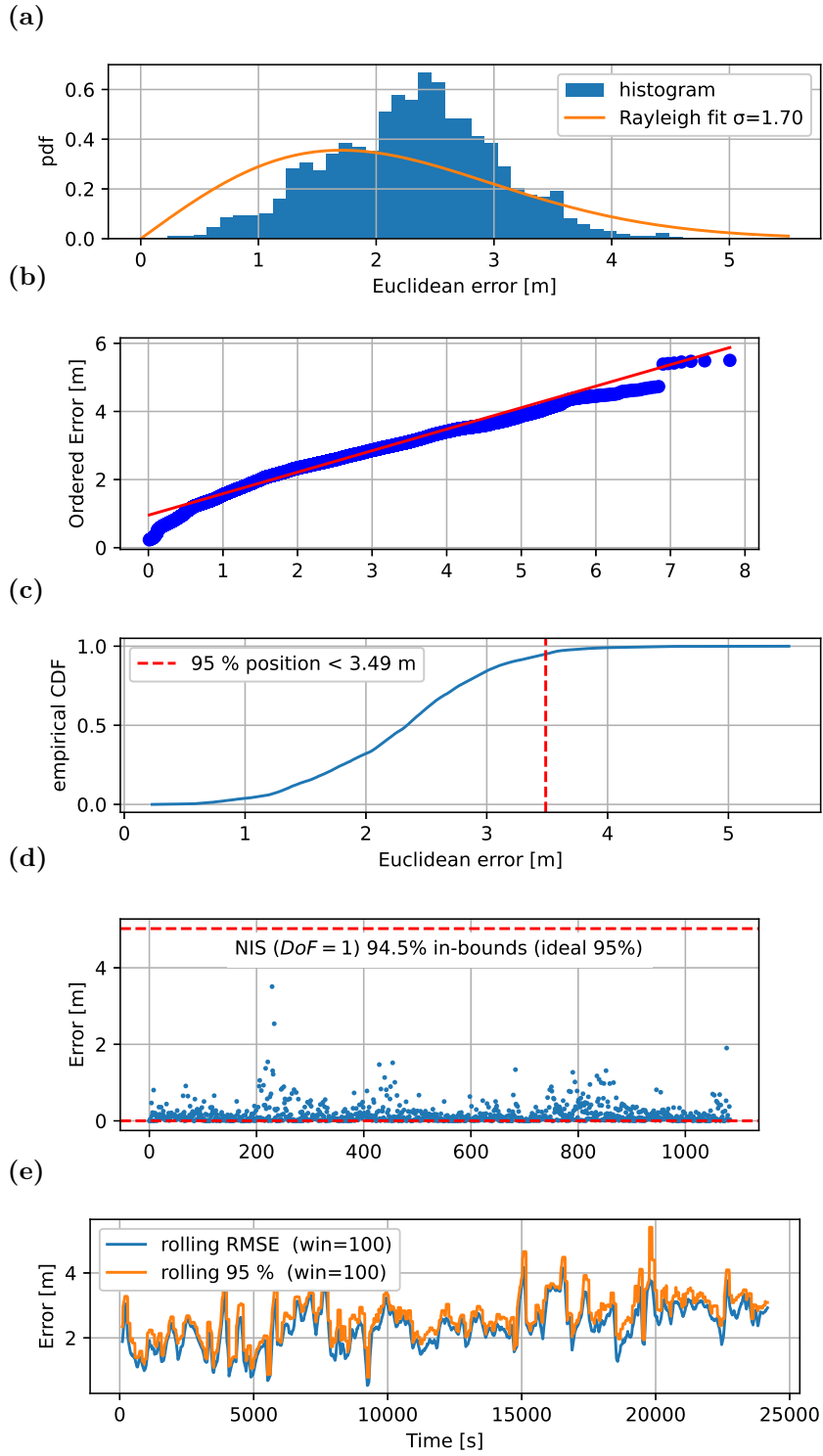


Figure 3.9. Diagnostic plots for EKF accuracy and consistency. (a) Histogram with Rayleigh fit; (b) Q-Q plot vs. Rayleigh; (c) Empirical CDF of ε_k ; (d) NIS consistency check; (e) Rolling RMSE and 95 %.

Lastly, the histogram in Figure 3.9a shows a broadened peak and a heavier tail, both consistent with a slowly varying range bias. Refining the EKF noise statistics about the beacon coordinates and range measures could allow the filter to better identify and absorb this bias.

After reviewing the state of the art on methods on the same class of the one presented in this Section, the thesis proceeds (in Section 3.3) in presenting an adaptive approach that tackles the points raised in this discussion.

3.2.4 Related Work

This section concerns works about the implementation and testing of methods for localizing AUVs based on LBL acoustic measurements, whose aims and purposes are similar to the work presented in this Section, namely, evaluating the localization error of the proposed method. These works can be broadly divided into those that are mainly based on LBL ranging like the presented one, and those who supplement these measures with information from sensors on board of the AUV.

Li et al. propose an LBL-based approach that achieves remarkable accuracy, with a mean localization error in the order of few centimeters [41]. The accuracy of their results stems from a long calibration phase and precise determination of the coordinates of the beacon, resulting in extremely accurate ranging. Their method also takes into account the different values of the speed of sound in water, depending on the depth of the AUV. The coordinates of the actual trajectory of the AUV are determined through Differential GPS (DGPS)³, which produces geodetic measurements that are within very few centimeters from the actual ones. This enables a very precise determination of the localization error.

Similarly, Han et al. obtain a mean localization error never to exceed 1 m [42] due to a ranging equipment that is capable of extremely accurate LBL measurements and to accurate GPS readings. These two methods achieve greater accuracy through the use of costly equipment that is not easy to deploy.

Among the works where the localization of an AUV is performed by supplementing ranging with measurements from sensors on board of the AUV, the method by Wang et al. [43] also includes heading measures from an IMU and velocity measures from a DVL. Their method, however, achieves a reported accuracy around 4 m, which is inferior to the results achieved in the presented work, obtained without using those additional measurements.

Sun et al. [44] develop a multi-beacon Unscented Kalman Filter (UKF)-based algorithm that combines ranges from four fixed beacons with a Doppler-velocity estimate inside an UKF to avoid the linearization errors of EKF formulations. Lake trials show that the UKF keeps the real-time horizontal (bounded by 5 m) error comparable to the one of the presented method, while their EKF has worse performance. Like the previous work, their method is also aided by an expensive DVL that injects the vehicle velocity into their formulation.

³Differential GPS is a technique used to improve the accuracy of LBL measurements by using a network of reference stations with known locations

Zhao et al. [45] propose a Range-Only Single-Beacon (ROSB) factor-graph fusion framework that augments range with INS, DVL and depth-sensor data. Their lake tests show that the resulting UMFP (Underwater Multi-sensor Fusion Positioning) keeps the planar RMS error below two metres while requiring only one beacon, thereby cutting deployment time relative to LBL. Their AUV, however, is supplemented with a DVL, a CTD and a compass, among other sensors, which make their approach much more complex than the method that this Section is concerned with.

Zhang et al. [46] present a tightly-coupled SINS/LBL filter that uses time-difference-of-arrival (TDOA) instead of absolute time-of-arrival to relax beacon synchronization. Simulation results report a mean position error of 1.54 m with four hydrophones, dropping to 1.19 m when a fifth hydrophone is added. However, the approach is only validated in simulation and the AUV model is equipped with a DVL and a magnetic compass.

In the work by Bernardi et al. a system is defined and implemented in support of divers and multimedia networking that uses the Zeno AUV and three beacons for localization [47]. The method uses LBL and the measurements from an on board IMU. Based on the reported results, the presented method outperforms theirs by producing an average localization error that is at least 80 cm smaller.

Other localization methods worth mentioning in this category of works includes those by Techy et al. [48], Wolbrecht et al. [49] and by Singh et al. [50]. These methods, however, do not evaluate the accuracy of the localization of the AUV, focusing instead on different metrics (such as the goodness of the trace of the covariance matrix of the filter they define) or present different setting not comparable to the ones presented in this thesis. As such, these approaches cannot be directly compared to the method in this section.

By way of conclusion, it is to know that while supplemental information from on board sensors might increase localization accuracy, their use could be prohibitive for certain AUVs, especially for those with small form factors, in terms of cost and payload. The distinguished contribution of the work presented in this Section is that of achieving high localization accuracy without requiring expensive payloads and costly and complex deployment or calibration operations.

3.3 An Adaptive Extended Kalman Filter for State and Parameter Estimation in AUV Localization

The previous Section has shown that a fixed-tuned EKF, fed only by depth and slant-range measurements, can already achieve metre-level accuracy in marine surveys. Its main weakness, however, resides in the the assumption that all the nuisance parameters, like the beacon coordinates and the stochastic properties of the sensors, remain stationary and known.

However field data reveal a different scenario: beacon moorings can be dropped with meter-scale uncertainty, acoustic multipath intermittently inflates range noise, and inertial sensors drift in ways that do not follow a single, time-invariant covariance. When these effects accumulate, filter performance can degrade. This Section builds upon the previously presented localization method, addressing the problem of non-

stationarity of the measurement noise, uncertainty on the process noise calibration and imperfect beacon surveying, devising a set of methods that can improve AUV localization. An adaptive EKF is introduced, that compensates for non-stationary phenomena *in situ*, without adding hardware or pre-deployment calibration time. In a nutshell, the proposed methodology is divided in two phases:

1. **GPS-aided Bootstrap.** During a short surface “warm-up” phase, GPS fixes and raw ranges are fused in a least-squares backend that yields centimeter-accurate beacon coordinates and associated covariances. Those covariances are later carried into the measurement model, so underwater updates remain consistent even if a buoy shifts by a few decimeters.

Additionally, successive GPS Course-Over-Ground (COG) fixes provide “snapshot” yaw measurements; their mismatch with the IMU-derived yaw is incorporated into the EKF and yields an empirical, data-driven initialization of the gyro-bias process noise.

2. **Inertial LBL Localization Phase.** After the GPS-aided phase, the navigation only relies on (i) IMU dead-reckoning and (ii) asynchronous LBL ranges. The process model is a 16-state inertial strap-down formulation that propagates position, quaternion attitude, velocity, and slowly varying IMU biases. Once submerged, a windowed statistics of the range innovations is used to update each beacon’s range variance $\sigma_{r,i}^2$ separately. Beacon-position covariances remain in the measurement model, accounting for the uncertainty inherited from the surface fit.

The rest of this Section is organized as follows.

- Section 3.3.1 details the GPS-aided start-up phase, including the least-squares beacon solver and the sliding-window update rule for the measurement covariance.
- Section 3.3.2 revisits the motion and observation models, showing how beacon states and noise hyper-parameters are injected into the classical EKF recursion.
- Section 3.3.3 performs ablation studies that isolate the contribution of each adaptive block in a simulated environment and evaluates the proposed method in real field tests at sea where the previously unknown coordinates of the beacons are estimated in about 3 minutes and the achieved RMSE localization accuracy is at meter level, obtaining a 53% improvement over a similar non-adaptive version used as baseline.
- Section 3.3.4 analyses the obtained results.
- Section 3.3.5 situates the approach within the literature on adaptive KF variants.

3.3.1 Beacon Positions and Noise Statistics Estimation

While the AUV is on the surface, the GPS and the pressure sensor give an absolute estimate of the vehicle pose $\mathbf{p}_k^G = [N_k \ E_k \ D_k]^\top$ at each time step k . Let d_k^i be

the acoustic slant-range measured from the vehicle to beacon i at the same epoch. Collecting all pairs $\{\mathbf{p}_k^G, d_k^i\}_{k=1}^K$, the unknown planar location of beacon i in NED, $\mathbf{p}^{b,i} = [x^{b,i}, y^{b,i}]^\top$, is obtained via a two-dimensional least-squares fit:

$$\hat{\mathbf{p}}^{b,i} = \arg \min_{\mathbf{p}} \sum_{k=1}^{K_i} \left(d_k^i - \|\mathbf{p} - \mathbf{p}_{k,\text{hor}}^G\|_2 \right)^2,$$

where $\mathbf{p}_{k,\text{hor}}^G = [N_k \ E_k]^\top$. At the solution $\hat{\mathbf{p}}^{b,i}$ the residuals

$$e_k = d_k^i - \|\hat{\mathbf{p}}^{b,i} - \mathbf{p}_{k,\text{hor}}^G\|$$

and the Jacobian rows

$$\mathbf{j}_k^\top = \frac{\hat{\mathbf{p}}^{b,i} - \mathbf{p}_{k,\text{hor}}^G}{\|\hat{\mathbf{p}}^{b,i} - \mathbf{p}_{k,\text{hor}}^G\|}$$

are evaluated and stacked into the matrix $J \in \mathbb{R}^{K_i \times 2}$. The per-range noise variance is estimated from the residuals as

$$\hat{\sigma}_r^2 = \frac{1}{K_i - 2} \sum_{k=1}^{K_i} e_k^2,$$

and the covariance of the beacon estimate is approximated by

$$\Sigma^{b,i} \approx \hat{\sigma}_r^2 (J^\top J)^{-1}.$$

The solution is accepted only if the root-mean-square residual

$$\text{RMS}_e = \sqrt{\frac{1}{K_i} \sum_{k=1}^{K_i} e_k^2}$$

does not exceed a user-defined tolerance τ [m], which accounts for the inaccuracies of the AUV's GPS.

As soon as at least $n \leq N$ beacons ($n = 3$ in the presented settings) meet this criterion, the vehicle leaves the surface and normal navigation begins; any extra beacon discovered later is processed in exactly the same way, but using the EKF position $\mathbf{p}_k$ instead of the GPS fix. Figure 3.10 shows the position estimation of three beacons deployed in a simulated scenario: as the AUV navigates it is able to determine the position of each beacon by fitting the intersection of the circumferences described by each of the computed ranges.

At run time the EKF uses the block-diagonal measurement-noise matrix

$$\mathbf{R}_k = \text{diag}(\mathbf{R}_k^0, \dots, \mathbf{R}_k^{N-1}), \quad \mathbf{R}_k^i = \text{diag}(\sigma_{r,i}^2, \sigma_{b,i}^2, \sigma_{b,i}^2),$$

where $\sigma_{r,i}^2$ is the variance of the current range reading and $\sigma_{b,i}^2$ the (fixed) variance of the beacon coordinates just computed. Splitting the full matrix into independent 3×3 tiles reduces the algebra in the update step and lets each beacon keep its own statistics.

To track slow changes in the acoustic channel, the variance of the range component is refreshed online:

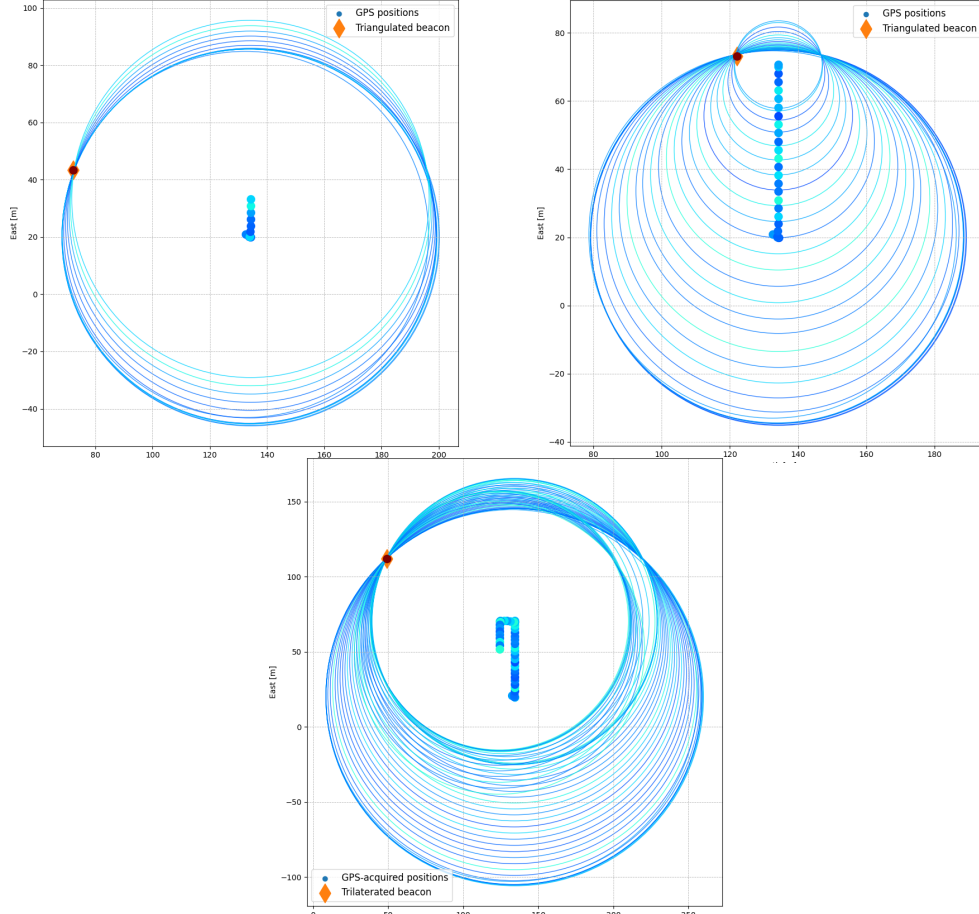


Figure 3.10. Position estimation of three deployed beacons (orange rhomboids) in a simulated scenario. During the AUV route (blue/cyan dots) the intersections of the circumferences described by the range towards each of the three beacons allow to determine their unique position.

1. For every beacon i whose range was received at step k , and the innovation $\rho_k^i = d_k^i - \|\hat{\mathbf{p}}_{k|k-1} - \hat{\mathbf{p}}^{b,i}\|_2$ is computed.
2. A sliding window $\rho_{k-m+1:k}^i$ of the last m innovations is kept and its sample variance $\sigma_k^{\rho,i}$ is evaluated.
3. (Heuristic) The variance $\sigma_k^{\rho,i}$ is down-weighted by the scalar Kalman Gain associated to the same measurement K_k^i so that the filter trusts noisy data less:

$$\sigma_{r,i}^2 \leftarrow K_k^i \sigma_k^{\rho,i}.$$

This adaptive scheme lets the EKF tighten the covariance when the innovations are small and loosen it as soon as noise or interference inflate the innovations, preserving consistency without the need for manual retuning.

3.3.2 System Model

At a discrete instant k , the EKF state is

$$\mathbf{x}_k = [\mathbf{p}_k^\top \mathbf{q}_k^\top \mathbf{v}_k^\top (\mathbf{b}_k^g)^\top (\mathbf{b}_k^a)^\top]^\top \in \mathbb{R}^{16},$$

where

- $\mathbf{p}_k = [x_k, y_k, z_k]^\top$ is the position in the NED frame;
- $\mathbf{q}_k = [q_k^w, q_k^x, q_k^y, q_k^z]^\top$ is the unit quaternion that rotates vectors from the *body* frame to NED;
- $\mathbf{v}_k = [v_k^N, v_k^E, v_k^D]^\top$ is the linear velocity in NED;
- $\mathbf{b}_k^g$ and $\mathbf{b}_k^a$ are the slowly-time-varying gyroscope and accelerometer biases expressed in the body frame.

Process Model. At a frequency $1/\Delta t$ ($\Delta t \simeq 0.1$ s) the AUV's IMU provides raw angular velocity $\boldsymbol{\omega}_m$ (rad/s) and specific force $\mathbf{a}_m$ (m/s²), corrupted by the slowly-varying biases shown in the state. With $\mathbf{g} = [0, 0, 9.80665]^\top$ and

$$\tilde{\boldsymbol{\omega}}_k = \boldsymbol{\omega}_{m,k} - \mathbf{b}_k^g, \quad \tilde{\mathbf{a}}_k = \mathbf{a}_{m,k} - \mathbf{b}_k^a.$$

The full state time propagation is

$$\begin{aligned} \mathbf{p}_{k+1} &= \mathbf{p}_k + \mathbf{v}_k \Delta t + \frac{1}{2} (\mathbf{R}(\mathbf{q}_{k+1}) \tilde{\mathbf{a}}_k + \mathbf{g}) \Delta t^2 \\ \mathbf{q}_{k+1} &= \mathbf{N}(\mathbf{q}_k + \frac{1}{2} \boldsymbol{\Omega}(\tilde{\boldsymbol{\omega}}_k) \mathbf{q}_k \Delta t) \\ \mathbf{v}_{k+1} &= \mathbf{v}_k + (\mathbf{R}(\mathbf{q}_{k+1}) \tilde{\mathbf{a}}_k + \mathbf{g}) \Delta t \\ \mathbf{b}_{k+1}^g &= \mathbf{b}_k^g + \mathbf{n}_k^{bg} \\ \mathbf{b}_{k+1}^a &= \mathbf{b}_k^a + \mathbf{n}_k^{ba} \end{aligned}$$

where $\mathbf{n}^{(\cdot)} \sim \mathcal{N}(\mathbf{0}, Q^{(\cdot)})$ are independent white-noises for the gyro and accelerometer biases and $\mathbf{N}$ normalizes the quaternion ($\mathbf{N}(\mathbf{q}_k) = \mathbf{q}_k / \|\mathbf{q}_k\|$). The $\boldsymbol{\Omega}(\boldsymbol{\omega})$ matrix translates angular rate into quaternion dynamics and $\mathbf{R}(\mathbf{q})$ is quaternion Direction Cosine Matrix (DCM) transforming vectors from body to world frames. Expanding them yields:

$$\begin{aligned} \boldsymbol{\Omega}(\boldsymbol{\omega}) &= \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \\ \mathbf{R}(\mathbf{q}) &= \begin{bmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & 1 - 2(q_x^2 + q_y^2) \end{bmatrix}. \end{aligned}$$

The discrete stochastic recursion is then:

$$\mathbf{x}_k = \mathbf{F} \mathbf{x}_{k-1} + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}),$$

with

$$\mathbf{F} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} & \mathbf{I}_3 \Delta t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_4 + \frac{1}{2} \boldsymbol{\Omega}(\tilde{\omega}) \Delta t & \mathbf{0} & -\frac{1}{2} \Xi(\mathbf{q}) \Delta t & \mathbf{0} \\ \mathbf{0} & [\partial R / \partial \mathbf{q} \tilde{\mathbf{a}}] \Delta t & \mathbf{I}_3 & -R(\mathbf{q}) \tilde{\mathbf{a}}^\times \Delta t & -R(\mathbf{q}) \Delta t \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{bmatrix},$$

$$\Xi(\mathbf{q}) = \begin{bmatrix} -q_x & -q_y & -q_z \\ q_w & -q_z & q_y \\ q_z & q_w & -q_x \\ -q_y & q_x & q_w \end{bmatrix},$$

$$\mathbf{Q} = \text{diag}(\sigma_\omega^2 \Delta t^2 \mathbf{I}_3, \sigma_a^2 \Delta t^2 \mathbf{I}_3, \sigma_{bg}^2 \Delta t \mathbf{I}_3, \sigma_{ba}^2 \Delta t \mathbf{I}_3) \in \mathbb{R}^{12 \times 12}$$

and $\tilde{\mathbf{a}}^\times$ is the skew-symmetric matrix of the measured (debiased) acceleration.

Measurement Models The presented formulation adopts three kinds of measurement models, embedded into the EKF:

- **GPS position. (only used during the GPS-aided phase)** As the name suggests, when at the surface, the AUV embeds directly the converted LLA coordinates measured from the on board GPS into the position components of the state directly, assuming them error-free ⁴. As such, this measurement model is trivial and is not formalized here;
- **Slant range measurement model.** The same shown in Section 3.2.1;
- **COG heading. (only used during the GPS-aided phase)** Given two consecutive GPS fixes (converted in the NED frame)

$$\Delta \mathbf{p}_k = \mathbf{p}_k^{\text{GPS}} - \mathbf{p}_{k-1}^{\text{GPS}} = [\Delta N_k \ \Delta E_k]^\top,$$

the COG heading is ⁵:

$$z_k^{\text{COG}} = \psi_k^{\text{GPS}} = \text{atan2}(\Delta E_k, \Delta N_k) \in (-\pi, \pi].$$

In the EKF state, attitude is encoded by the unit quaternion $\mathbf{q}_k = [q_k^w, q_k^x, q_k^y, q_k^z]^\top$. The yaw angle extracted from $\mathbf{q}_k$ is

$$\hat{\psi}_k = h^{\text{COG}}(\mathbf{x}_k) = \text{atan2}(2(q_k^w q_k^z + q_k^x q_k^y), 1 - 2((q_k^y)^2 + (q_k^z)^2)).$$

⁴GPS coordinates are assumed error-free because we assume, in this Chapter, that the on board GPS is the most accurate (position-wise) sensor among those in use. The aim of this assumption is uniforming all the measurements (e.g., beacon coordinates and relative ranges, as will be seen further in this Section) in the frame of the onboard GPS, that is used as Ground Truth during the evaluation.

⁵ $\|\Delta \mathbf{p}_k\| > d_{\min}$ is enforced to avoid numerical problems when the vehicle does not move between fixes.

Defining $s_k = 2(q_k^w q_k^z + q_k^x q_k^y)$, $c_k = 1 - 2((q_k^y)^2 + (q_k^z)^2)$, the Jacobian row that enters the EKF update is

$$\mathbf{H}_k^{\text{cog}} = [\mathbf{0}_{1 \times 3} \quad \partial \hat{\psi}_k / \partial \mathbf{q}_k \quad \mathbf{0}_{1 \times 10}],$$

with

$$\frac{\partial \hat{\psi}_k}{\partial \mathbf{q}_k} = \frac{1}{c_k^2 + s_k^2} \begin{bmatrix} 2q_k^z & 2q_k^y & -4q_k^y q_k^z & 2q_k^w - 4(q_k^z)^2 \end{bmatrix}.$$

When the Update step of the EKF is triggered by a GPS COG measurement, the innovation, Kalman gain and correction follow the standard EKF rules with the scalar residual $y_k^{\text{cog}} = z_k^{\text{cog}} - h^{\text{cog}}(\mathbf{x}_k)$, Jacobian $\mathbf{H}_k^{\text{cog}}$, and variance $\mathbf{R}_k^{\text{cog}}$.

What this means is each GPS-based COG reading acts like a short “snapshot” of the true yaw that is absolute and not subject to gyro drift. By comparing this trusted angle to the yaw predicted from the gyros, the EKF can treat any mismatch as evidence of yaw-specific noise.

The classical predict and update steps of the EKF recursion are the same as shown in Section 3.2.1.

3.3.3 Experimental Evaluation

To evaluate the presented method, two different sets of experiments using synthetically simulated and real data from the scenario presented in Section 3.1 have been performed.

The aim of the simulations is that of individually validating in lab the effectiveness of the adaptive techniques and models presented in the previous Section. Simulations are carried out as an ablation study, in which the effects of each optimization block is evaluated separately and then evaluated in terms of localization error.

The accuracy of the overall method is finally evaluated on data collected in actual measurement campaigns at sea [51]. The localization error (RMSE) between the estimated location and a GPS ground truth is used as a main mean of evaluation, along with the other accuracy and stability metrics that have also been used in the previously presented "baseline" method (in Section 3.2.2).

Although in different settings, both simulation and field experiments share a similar operational mode, which is described in the following. As soon as the beacons and the AUV are deployed, the AUV immediately enters the GPS bootstrap mode, in which the beacon position estimation and range measurement noise characterization (described in Section 3.3.1) is carried out. In this phase, the EKF operates using the additional GPS position and COG heading measurement models, in addition to the canonical slant range measurement model, which is always active. After the GPS-aided phase, the AUV switches back to the classical EKF recursion, computing the Predict step each time a measure from the IMU is received and polling the beacons to compute ranges, triggering the Update step.

Table 3.4 shows the values of the threshold on beacon position accuracy τ , the size m of the sliding window for range noise estimation, and the initial values of the process and measurement noise covariance matrices, for both simulations and tests at sea. This Section now proceeds in showing, in two separate paragraphs, the results obtained from the simulation studies and field test at sea, respectively.

Quantity	Sym.	Unit	Simulation	Field test
Accuracy threshold	τ	m	0.4	0.4
Window length	m	—	20	20
Gyro process variance	σ_ω^2	rad ² /s ²	0.01	0.10
Accel. process variance	σ_g^2	m ² /s ²	0.01	0.10
Gyro-bias variance	σ_{bg}^2	rad ²	0.01	0.10
Accel.bias variance	σ_{ba}^2	m ² /s ⁴	0.01	0.10
Beacon-pos. variance	σ_b^2	m ²	1.20	1.20
Range-meas. variance	σ_r^2	m ²	1.20	1.20

Table 3.4. Initial EKF state and covariance parameters used in simulations and field experiments.

Simulation Results

The simulated scenario consists of three beacons and an AUV moving on the surface according to a lawnmower path at a median speed of about 0.2 m/s, in an area of about 30 m × 60 m. We evaluate each of the proposed optimization techniques in such a scenario.

1. Unknown Beacon Coordinates Estimation. The AUV’s belief on the location of each beacon has been artificially shifted 5 m North-West. For the first t seconds, the AUV gets its position using GPS measurements arriving at 10 Hz. Its position at time k will then be $\mathbf{p}_k = \mathbf{p}_k^{gps}$, $\forall k \leq t$. During this phase, the beacon position estimation procedure described in Section 3.3.1 is carried out. In the considered scenario, this method can successfully localize all the beacons after about $t = 200s$.

Figure 3.11 shows the estimated trajectory of the AUV after the completion of the beacon localization phase, annotated with the estimated beacon locations (in green), versus the trajectory (light blue) estimated by the same filter with biased beacon location beliefs. As can be seen from the trajectory plot, the trajectory estimate that relies on biased beacon coordinates is significantly shifted with respect to the ground truth, while a proper beacon location belief ensures a correct trajectory estimation (black line).

2. Adaptive Process Noise Covariance In this experiment, the orientation part of the EKF state $\mathbf{q}_k = [q_k^w, q_k^x, q_k^y, q_k^z]^\top$ is artificially perturbed by a random noise $\delta\psi \sim \mathcal{N}(0, 0.2^2)$ during the predict step. At each time k , $\mathbf{q}_k$ becomes:

$$\begin{aligned}\mathbf{q}_k^{biased} &= \mathbf{q}_k \otimes \mathbf{q}_{noise} \\ \mathbf{q}_{noise} &= [0, 0, \sin(\delta\psi/2), \cos(\delta\psi/2)]\end{aligned}$$

At the end of the GPS-aided phase, the covariance σ_{cog} of the collected residuals $\mathbf{y}^{cog} = y_k^{cog} \forall k \leq t$ is computed as a proxy estimate of the noise of the AUV heading⁶. The residuals are not collected in quaternion form, but they are converted

⁶Although incorporated in the presented model, only the heading and the planar motion of the AUV are considered here, as, as seen later in this Section, the observed tilt on the roll and pitch directions of the vehicle are negligible to the specific purpose of this method.

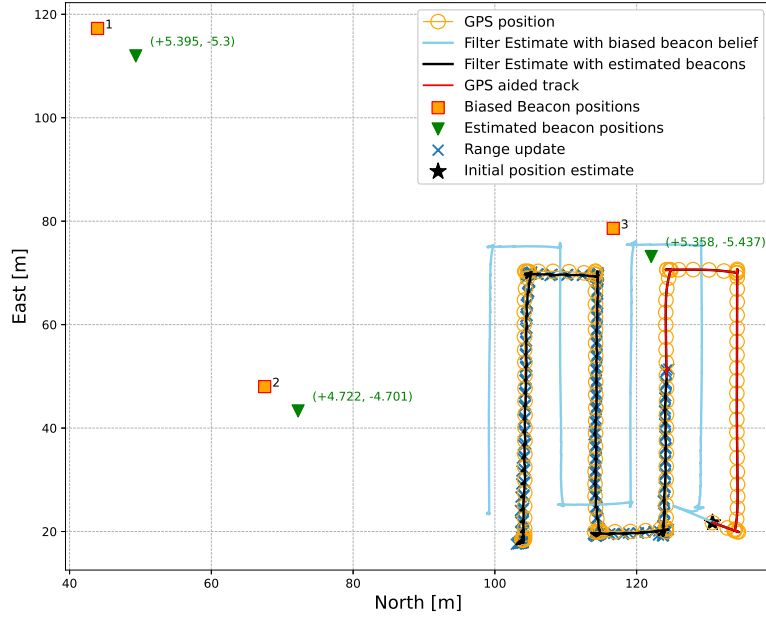


Figure 3.11. Estimated trajectory after beacon localization (black solid) vs. estimated trajectory with biased beliefs about the beacon locations (light blue).

into a Roll(ϕ)-Pitch(θ)-Yaw(ψ) representation ⁷ as (the k subscript is dropped for readability):

$$\phi = \text{atan2}\left(2(q_y q_z + q_w q_x), 1 - 2(q_x^2 + q_y^2)\right),$$

$$\theta = \arcsin(2(q_w q_y - q_x q_z)),$$

$$\psi = \text{atan2}\left(2(q_w q_z + q_x q_y), 1 - 2(q_y^2 + q_z^2)\right).$$

Figure 3.12 shows the comparison between the artificially induced Gaussian noise on the heading measures and a population, with same sample size, drawn from $\sim \mathcal{N}(0, \sigma_{cog}^2)$. The estimated additive noise on yaw measures throughout the GPS-aided phase (yellow) closely follows the distribution of the artificial noise induced in simulation (green). In fact, the estimated value of σ_{cog} (0.18 rad^2) is very close to that of the artificially induced noise on the yaw measures (0.2 rad^2). Consequently, it can be concluded that σ_{cog}^2 is a valid proxy for the noise that the yaw estimates underwent during the GPS-aided phase in this experiment.⁸

3. Adaptive Measurement Noise Covariance In this experiment, after the GPS-aided phase, each slant range measured by the AUV through acoustic ranging has been artificially perturbed with additive noise following a Half-Normal

⁷This representation is not needed for the sake of state estimation but it is chosen to better visualize and interpret the analysis and plots regarding orientation, as the quaternion form is not easily readable.

⁸In case of non-negligible water dynamics effects (e.g., currents), this method would incorporate them in σ_{cog}^2 . In that case, estimating water dynamics would help in ruling out noises that are independent of the instrument.

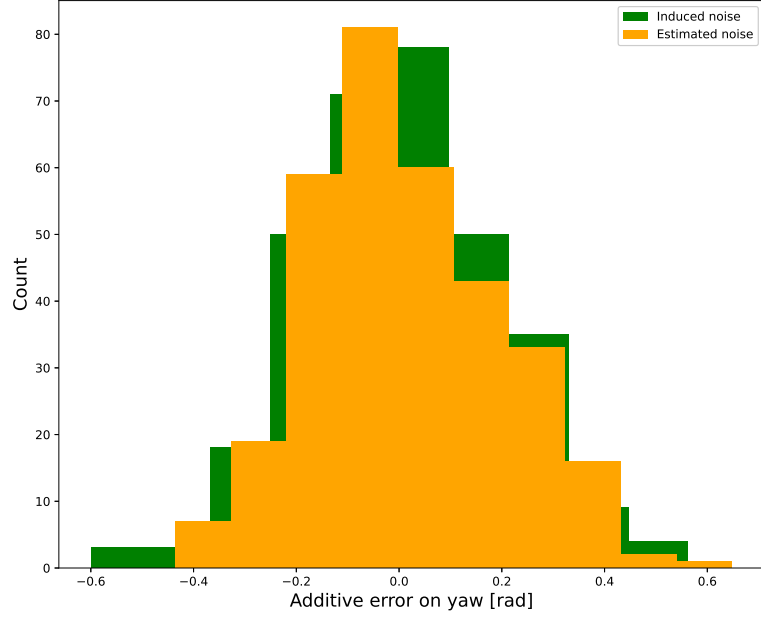


Figure 3.12. Comparison between the artificially induced Gaussian noise perturbing the yaw measures (green) and the noise estimated by our method (orange).

distribution $\eta_r \sim \mathcal{HN}(0, \sigma_{noise,r}^2)$. This kind of induced noise has been selected to reflect the fact that the only range-related noise that could affect the localization of the AUV, once the beacons have been localized, is that affecting the range measures themselves, *i.e.*, r_r . Specifically, assuming the beacons do not move around more than τm ,⁹ a range measure can never be lower than the direct path we are trying to estimate. In fact, while multipath and reflections can cause ranges to overestimate the actual distance, the ToF used to compute them can never be lower than the SSP-induced propagation time.

Figure 3.13 shows the comparison between the artificially induced half-normal noise and the distribution of the residuals $\mathbf{y}_{b_i}$ accumulated during the whole mission and estimated by the presented method, relative to each of the involved beacons b_i . The estimated residuals on range measures throughout the mission, closely follows the distribution of the artificial noise that has been induced in simulation for all the beacons, confirming the effectiveness of the proposed solution in the considered scenario.

Table 3.5 summarizes, for each of the three deployed beacons, the covariance $\sigma_{noise,r}^2$ of the induced additive perturbation and the covariance on the range measures σ_r^2 at each sliding window update.

Sea Trials Results

As for the method presented in Section 3.2, the localization accuracy of this approach is computed by comparing the trajectory estimated by the presented EKF formulation and that measured by the on board GPS, showing their RMSE. The proposed

⁹ If they do, σ_r^2 would also be affected by the uncertainty that should be attributed to σ_b^2 . However, the noise estimations would still ensure proper adaptation.

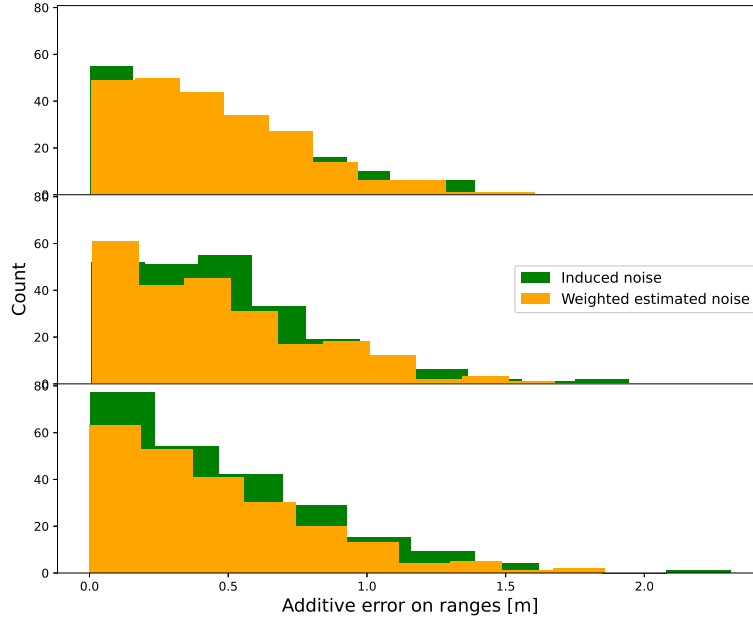


Figure 3.13. Comparison between the artificially induced half-normal noise perturbing the range measures (green) and the residuals estimated by the presented method (orange) on each beacon (top to bottom).

Beacon	Injected $\sigma_{noise,r}^2$	Estimated σ_r^2
B ₁	0.26 m ²	0.24 m ²
B ₂	0.31 m ²	0.29 m ²
B ₃	0.36 m ²	0.34 m ²

Table 3.5. Estimated σ_r^2 versus injected $\sigma_{noise,r}^2$ range noise in the simulated scenario.

localization method is then compared against the approach presented in Section 3.2, in which the AUV only knows the biased beacon locations measured with an imprecise GPS and adopts a fixed initial belief of the model noise statistics.

Figure 3.14 shows the 2D estimated AUV trajectory versus the one estimated by the GPS and the computed Euclidean localization error in time, along with the RMSE and the adjusted beacon coordinates estimated during the GPS bootstrap. The proposed method allows the AUV to self-localize with an error of 1.18 m, a 53% improvement over the EKF presented in Section 3.2. As the current method includes the initial GPS-aided phase, the computation of the localization error does not include the transient state, *i.e.*, the data collected during the first 3 minutes (time needed to localize the first 3 beacons), for fair comparison against the baseline, whose first 3 minutes of trajectory were therefore not considered.

Figure 3.15 shows the distribution of the estimated range residuals computed from the technique shown in Section 3.3.1, after the GPS-aided phase. Their windowed sample covariance is used to update the range noise covariance term in the measurement covariance matrix.

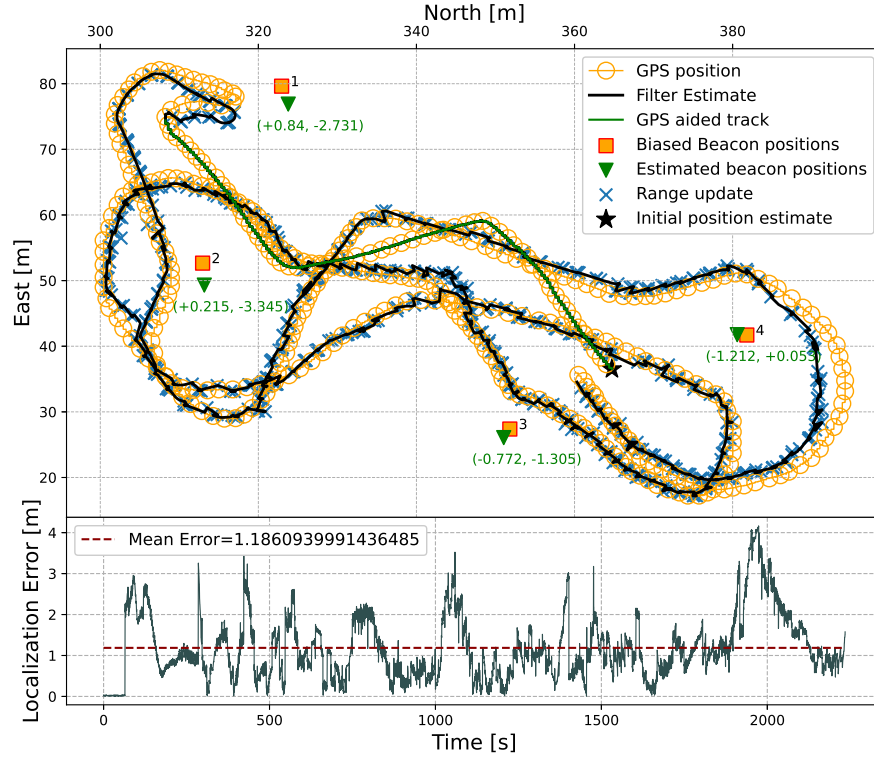


Figure 3.14. 2D Trajectory estimated by the proposed method and the resulting localization error. The green upside-down triangles show the adjust beacon coordinates estimated during the GPS-aided phase.

The AUV velocity estimation is analogous to the one shown in Section 3.2.2, and is therefore not shown here.

Figures 3.16 and 3.17 show the vehicle orientation (in Roll, Pitch, Yaw conversion) estimated during the mission. As can be seen, the AUV maneuvers and the experiment settings have been such that the vehicle barely underwent motion on these two directions.

In Figure 3.18 the statistical analysis carried out on the accuracy and consistency of the presented adaptive EKF, is shown. Figure 3.9c shows the empirical CDF of the 3D Euclidean position error $\varepsilon_k = \|\hat{\mathbf{x}}_{k|k} - \mathbf{x}_k^{\text{truth}}\|$. The 95th percentile occurs at 2.5 m, hence the majority of localization error sit fairly below that, being the RMSE over the entire run 1.18 m.

Consistency checks in the innovation space yield the NIS in Figure 3.18d, reporting that 94.7 % of the samples fall inside the two-sided 95 % χ_1^2 envelope.

Figure 3.18a shows that, the maximum-likelihood Rayleigh curve with $\sigma = 1.02$ m almost coincides with the empirical bars and the tail up to ~ 4 m is accurately captured, indicating that the planar innovations behave like two independent $N(0, \sigma^2)$ variables, as assumed by the filter model.

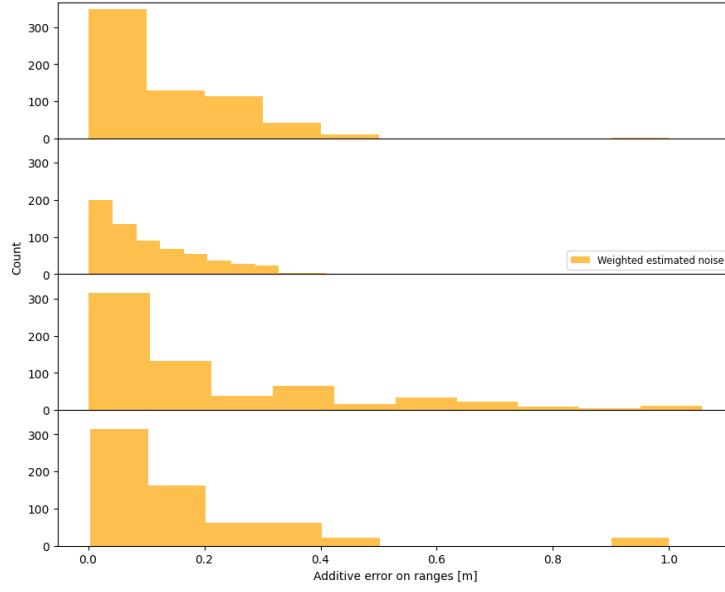


Figure 3.15. Range residuals estimated by the presented method (orange) on each beacon (top to bottom) in the sea trials.

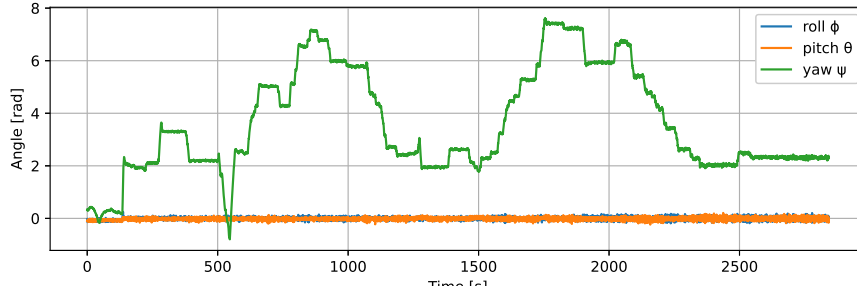


Figure 3.16. (In radians - unwrapped) Roll, Pitch and Yaw angles estimated during the mission.

The Q-Q plot (Figure 3.18b) ordered sample quantiles lie on the red 45° line until roughly the 97th percentile. falling a little below it, revealing a marginally lighter tail.

Figure 3.18e shows an RMSE closely following the bounds, not exhibiting pessimism nor optimism in the tuning process of the filter’s noise.

Finally, the State Covariance Matrix $\mathbf{P}$ stayed positive definite for the whole duration of the experiment.

In the next Section, the presented results will be discussed and a comparison against the baseline approach is performed to assess the proposed adaptive estimation techniques formally.

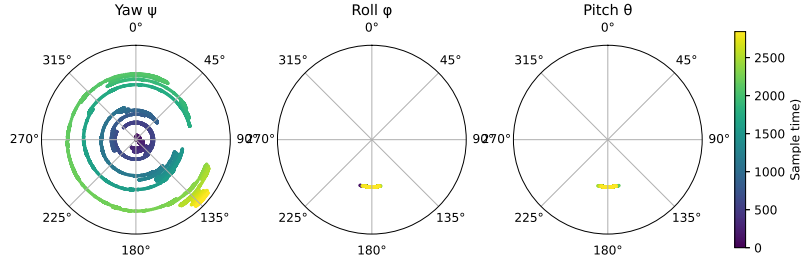


Figure 3.17. (In degrees - polar coordinates) Roll, Pitch and Yaw angles estimated during the mission.

3.3.4 Results Discussion

In the baseline invariant EKF the histogram in Figure 3.9a shows a noticeable full meter shift with respect to the Rayleigh mode and an understated tail above 4 m, raising the fitted scale parameter to $\sigma = 1.70$ m. The smaller ($\sigma = 1.02$) obtained by the proposed adaptive solution and the Rayleigh fit tracking the empirical histogram almost perfectly (3.18a), signal a smaller residual cloud after proper beacon bias removal and $\mathbf{R}$ tuning.

The "S-shape" in the Q-Q plot in 3.9b, revealed the presence of too few very small errors, an excess of mid-sized errors, and an even larger excess in the upper tail, possibly reflecting both a systematic bias and mis-specified noise statistics. However, it becomes straight up until the 97th percentile in the adaptive version (even if large errors still appear slightly overestimated) showing that filter's covariance correctly characterizes the error.

Even though 94.5 % of the NIS (Figure 3.9d) samples fall below the $\chi^2_{0.95}$ bound in the baseline filter, they include frequent excursions, signaling a possible subtle but systematic under-estimation of the measurement noise R . While both the models pass the χ^2 test, the presented adaptive formulation shows a slightly smaller number of innovation bursts (peaks).

Overall, the presented adaptive methodology, yielded a localization error of only 1.18 m, the ability of performing self-calibration operations without human intervention and a potential in adapting to time-varying noise conditions during navigation, while staying statistically consistent.

The next Section reviews approaches in literature that present adaptive methods of the same class of the presented one, and concludes this Chapter.

3.3.5 Related Work

Several works exist in the literature on adaptive solutions to tackle the time-varying noise characteristics of both terrestrial [52–55] and underwater [56–63] dynamic systems.

Among those concerning AUV localization in the underwater domain, the work by Wang et al. proposes an adaptive Sage-Husa-based Unscented KF (UKF) that allows

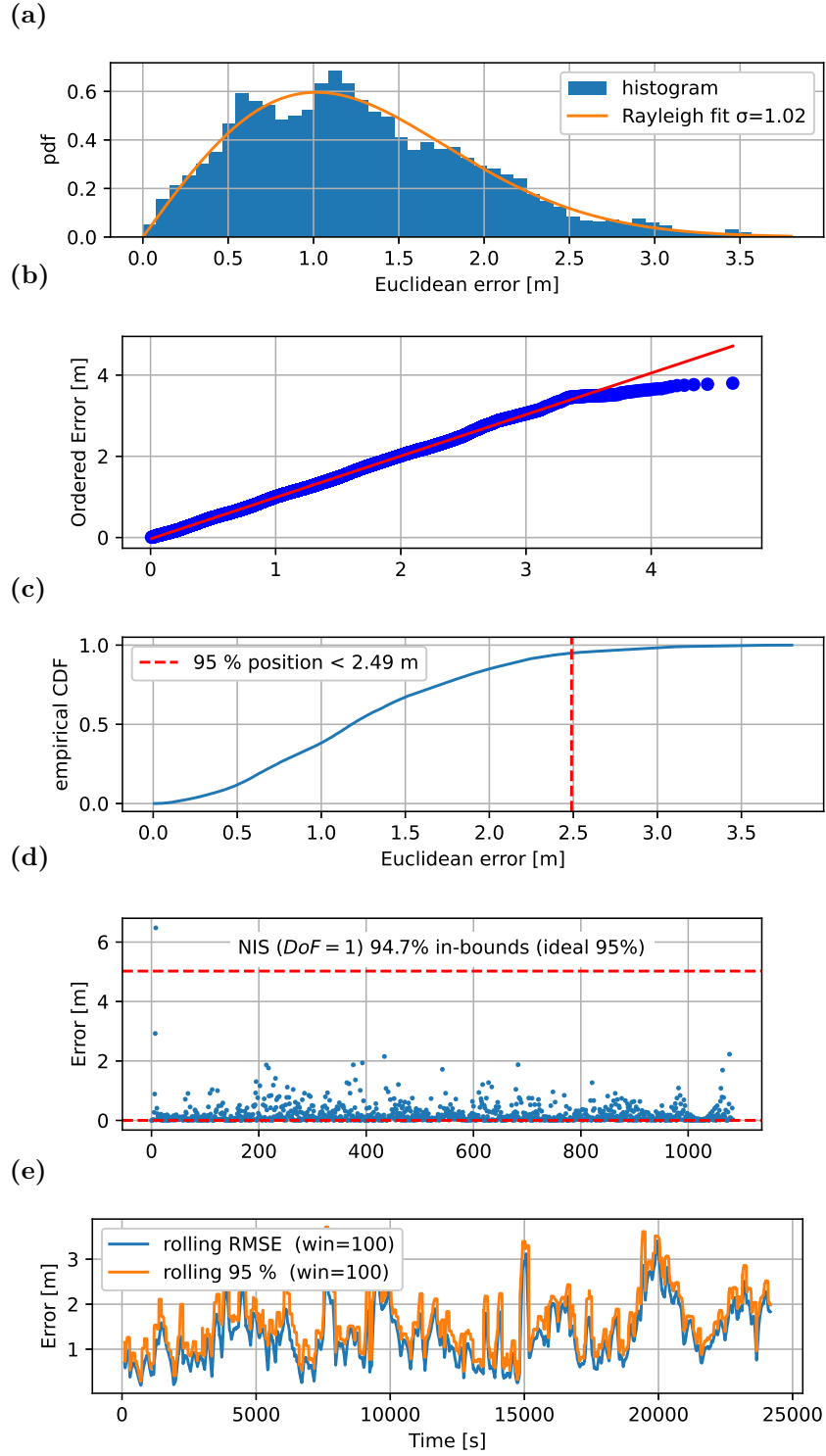


Figure 3.18. Diagnostic plots for the adaptive EKF accuracy and consistency. (a) Histogram with Rayleigh fit; (b) Q-Q plot vs. Rayleigh; (c) Empirical CDF of ε_k ; (d) NIS consistency check; (e) Rolling RMSE and 95 %.

estimating the process noise and measurement covariance matrices [56]. Instead of using an AUV, the method is validated by localizing a body that is towed by a vessel equipped with a costly suite of navigation sensors. While they report sub-meter Root Mean Square Error (RMSE) their setting is quite different from the one treated in this Section, which is concerned with actual low-cost AUVs and instruments.

Wang et al. [61] merge pseudo-range acoustic measurements with IMU and DVL data inside an EKF where the measurement-error statistics are updated on-line from the innovation residuals. The authors report a localization error bounded by 10 m. This Section's method greatly outperforms it, without the use of a DVL.

Zhao et al. [63] tackle single-beacon navigation with an Adaptive Network Fuzzy Inference System (ANFIS)-tuned EKF that re-estimates the state-transition model at every step. Field tests on a 100 m run show that the ANFIS-EKF lowers the 2-D positioning error error to ~ 6.2 m, versus 10.3 m for a conventional EKF, and ~ 6.8 m for a stand-alone ANFIS estimator. While clearly better than a fixed-gain EKF, this accuracy is still larger than the RMSE obtained in the sea trials presented in this Section without the need for neural-network tuning.

Xu et al. [57] propose an Ultra-Short Baseline (USBL) positioning system based on adaptive Kalman Filtering to adjust the process noise and the measurement covariance matrices using the residuals of the latest measurement data. An adaptive method is suggested by Shao et al. [58] to adapt the process noise and measurement covariance matrices of an EKF and an AUV performing dead reckoning using measurements from a Doppler Velocity Log (DVL) and from an Attitude and Heading Reference Systems (AHRS). The comparison between the GPS ground truth and the filter estimates of these two works suggests a localization error higher than that of the presented approach. However, no localization error metric has been reported by the authors, which makes a comparison with their work difficult.

In order to adapt the process noise covariance matrix to possibly varying process noise, Hajiyeve et al. [59] propose a UKF in which an adaptive process noise scale factor is put into the covariance matrix of the innovation sequence. Their method is evaluated via simulations. As the authors do not apply adaptive strategies to the measurement covariance matrix and do not provide experiments at sea, a comparison would be scarcely informative.

Sun et al. [60] propose a modified version of a Sage-Husa filter with an adaptive forgetting factor for an AUV using dead reckoning navigation. They test their solution at sea, reporting a localization error up to 9 m, which is improved in this work.

Estimating unknown beacon coordinates is performed extensively in terrestrial applications and in Simultaneous Localization and Mapping (SLAM) [64]. It has also been attempted in the context of underwater localization. Vaganai et al. [65], for instance, use the *Least Squares* method to home an AUV to a single beacon of unknown coordinates. Instead of using GPS (as this work does), the AUV uses dead reckoning while diving in a circular motion to disambiguate possibly collinear positions. Such approach is not directly comparable to ours, as no tests on localization accuracy are presented.

Olson et al. [66] use a voting scheme to determine the most likely beacon coordinates after filtering out outliers in range measurements. They also employ

a gradient-based method to maximize the efficiency of the AUV trajectory to disambiguate as many beacon position estimates as possible. They test their approach against a baseline EKF with knowledge of the beacon positions. The work concerns only beacon localization, and does not report on the accuracy of localizing AUVs. While they do not provide any metrics on the accuracy of AUV localization, they report that their beacon localization strategy yields translation errors between 1.85 m and 2.89 m, which this work improves.

Zhang et al. [62] introduce an Equivalent Virtual LBL (EVLBL) model coupled with particle-swarm optimisation to jointly calibrate all beacon coordinates in a single pass. Simulations demonstrate that the proposed EVPSO algorithm brings the beacon-position error down from 6.4 m to ≤ 1 m and, after compensation, reduces the LBL positioning error from 5 m to ~ 1 m. However, the approach is only evaluated in simulations, and the AUV is equipped with a DVL and magnetic compass, while the presented method reaches a similar accuracy without using these sensors.

Teixeira et al. [67] propose an EKF-based estimation technique aimed at maximizing the determinant of the innovation covariance matrix to determine the most likely beacon coordinates. The authors do not provide direct localization accuracy metrics for comparison but they plot the mean euclidean error in time. It can be seen that, using an optimized AUV trajectory, they are able to converge to a localization error comparable to the one achieved by this work. However, their method is only tested by simulations and requires initial estimates of the beacon positions.

Newman et al. [68] propose a range-only based scheme to enable an AUV to determine the coordinates of the beacons while moving. Their approach is tested against an EKF with knowledge of the true beacon positions. The authors provide error metrics concerning the lengths of the respective *inter-beacon* baselines (comprised between 3.7 m and 10 m) and a plot of the estimated vs. EKF “ground-truth” trajectories, which suggests that their localization accuracy is lower than the one of the presented work. The latter two works consider the problem as a SLAM problem, i.e., they include the beacon coordinates in the state of the EKF.

Chapter 4

MAREA: a Multimodal Co-Simulation Framework for Marine Networked Agents

Designing solutions for the underwater realm entails overcoming a series of challenges on logistics and monetary fronts prior to the ones posed by scientific or industrial objectives for which underwater robots have been increasingly employed. Moreover, flexibility and fault tolerance are both necessary in initial deployments due to the limited room for error while conducting experiments off-shore. Robots are in fact heavily used in underwater applications that would otherwise be dangerous to humans, *e.g.* requiring multiple consecutive immersions in cold waters, or impossible due to strength or endurance requirements. Nonetheless, this approach comes with the often non-negligible costs of robots and communication modems. On the latter point, finding the most promising set of modems for a robot, considering the pace of advances in communication technologies (acoustic, wireless optical, or electromagnetic), could result in too frequent workshop operations and prolonged mission delays.

In order to allow fast iteration during prototyping and tuning, simulators have been invaluable tools for research and development tasks in both robotics [69–72] and networking communities [73–75]. The underwater community often leverages specific simulators that take into account the unique conditions posed by the underwater environment. SUNSET [76], ns-3’s UAN Framework [77], DESERT [78], and AquaSim-NG [79] are among the most known frameworks to handle communication in the water medium. Regarding robotics and Cyber-Physical Systems (CPSs), multiple simulators exist that allow the accurate modeling of the physical properties of water and its effects on simulated entities. Among those that are suitable for multi-robot simulation in submerged environments, UNav-Sim [80], UWSim [81], UUV Simulator [82], DAVE [83], MARUS [84] are the most commonly known frameworks.

While simulators providing a comprehensive support for both realistic physics and communication exist in terrestrial or aerial domains, a similar tool is lacking within the underwater community. The work presented in this chapter aims to fill such gap in the literature presenting, at the time of writing, the first co-simulation environment for multi-robot scenarios with wireless multimodal networking capabilities.

Summarizing the contributions, the proposed architecture provides:

1. A containerized solution for simulating collaborative multi-robotics experiments, leveraging realistic underwater physics and a full-fledged control of networking stack features;
2. A modular architecture bridging popular open-source projects such as GAZEBO/UUV Simulator and ns-3 simulator, in order to drive adoption among both underwater robotics and networking communities;
3. A synchronization protocol for inter-simulator communication via raw sockets for consistent timing and message exchange across simulation timelines, sufficiently flexible for the integration of other simulators;
4. An ns-3 framework for underwater wireless multimodal communication, implementing several technologies (acoustic, optical, and electromagnetic modems), multiple networking stack protocols (at Medium Access Control and routing levels), and commonly used network performance metrics.

The rest of this chapter is organized as follows. Previous work on co-simulation environments in the aerial and ground domain, and details on the state of the art in underwater network simulation is given in Section 4.1. In Section 4.2 the design of the MAREA framework is described. The evaluation and benchmark of the proposed approach is found in Section 4.3. Finally, conclusions are drawn in 4.4.

4.1 Related Work

The development of co-simulation frameworks integrating robotic and network simulators has been an active research area, aiming to bridge the gap between CPS and networked robotic applications. Several frameworks have been proposed to enable realistic and synchronized co-simulation environments, each addressing specific challenges such as time synchronization, middleware interoperability, and scalability.

In this section, an overview of the most relevant works is provided, highlighting their intended field of application. While, at the time of writing, no previous solution is found in literature that provides a comprehensive co-simulation environment for networked underwater agents, several works exist that integrate network and physics-based simulations in the aerial and ground vehicles domain.

One of the earliest comprehensive co-simulation frameworks, CPS-Sim [85], integrates MATLAB/Simulink for physical system modeling and QualNet (or OM-NeT++ [73]) as the communication network simulator. The framework is primarily designed for wireless sensor networks, focusing on time synchronization challenges between periodic and event-driven simulators.

Among the frameworks that have emerged to integrate robotic and network simulators with a particular focus on multi-agent systems, the ROS-NetSim framework [86] was developed to simulate Perception-Action-Communication (PAC) loops in multi-robot systems. It is lightweight and modular, allowing integration with different network and physics simulators. Similarly, CORNET [87] integrates GAZEBO

with ns-3 to provide APIs for real-time synchronization and migration of control algorithms to physical platforms. Both of the latter two frameworks focus on enabling co-simulation for UAV networks. Specific efforts have been made to enhance the realism of UAV communication and control. FlyNetSim [88], for instance, synchronizes UAV flight dynamics (using ArduPilot) with network simulations in ns-3. Its architecture allows for real-time emulation of UAV-network interactions, making it suitable for urban IoT applications. In a similar fashion, the work in [89] also leverages GAZEBO, Arducopter and ns-3 to provide a software middleware to build an architecture for the realistic simulation of multi-UAV applications. To address the need for heterogeneous multi-agent co-simulation, SynchroSim [90] proposes an integrated middleware that dynamically adjusts time synchronization using a velocity-aware windowing approach in networks of UAVs and Unmanned Ground Vehicles (UGV). A more recent work [91], built upon SynchroSim, extends its capabilities by leveraging ROS 2's real-time capabilities and publish-subscribe transport for enhanced synchronization.

A noteworthy effort to achieving realistic underwater robotics simulations targeting both mobility and acoustic-based communication is UWRange [92]. This work integrates an ad-hoc packet transmission simulator, but it is uniquely designed for acoustic ranging, disregarding general-purpose communication-based applications that demand a more thorough protocol stack implementation.

Among networking simulators, ns-3 [74] is a widely used open-source network simulator popular among both industrial and academic researchers, and it is still actively developed. It models underwater networking, providing acoustic communication support via its integrated UAN Framework [77], lacking however native support for a number of features that are important for tackling complex underwater scenarios. Nowadays, features such as a unified approach to routing protocol implementation, multiple reconfigurable protocol stacks, and communication implemented via technologies different from acoustic are highly requested, yet the UAN Framework has not been updated to reflect these trends.

SUNSET [76] and DESERT [78] are underwater network simulators, emulators, and real-time testing frameworks originally both based on ns-2 [93] and NS-MIRACLE [94]. While SUNSET has since been significantly extended becoming a very powerful simulation and emulation tool, it has also become proprietary software. DESERT has also undergone extensive development since its first release, and both software suites now provide support to multistack and multimodal underwater networks. However, the dependency from ns-2 is a drawback, as the community has since moved away from this simulator given its unreliable memory management.

AquaSim-NG [79] has been developed as a porting from ns-2 to ns-3. This framework overcomes the standardized routing implementation issue, but remains an exclusively single-stack and acoustic-only simulator and it is not compatible with ns-3's own UAN Framework and its upper layers. Additionally, none of the mentioned network simulators provide mobility support beyond simplified models.

Overall, integrating robotics and network simulators is a promising path to achieve the realization of complex objectives in the underwater domain, where multi-agent applications are on the rise. Without doing so, physical simulators would ignore the effects of communication channels and disregard the nuances

introduced by a full-fledged protocol stack. Specularly, network simulators usually do only provide simple mobility modeling, as distances and relative angles between agents generally suffice in their primary focus of channel modeling; by ignoring physical agent dynamics and mobility, many network simulators omit key factors such as realistic motion constraints and neglect possibly more sophisticated and context-dependent navigation decisions.

4.2 MAREA Framework Design

The proposed MAREA co-simulation framework is composed of three main modules:

- **MAREA-CP**, the Cyber-Physical simulator, an extension of UUV Simulator [82].
- **MAREA-ns3**, the networking framework for multimodal and multistack UWSNs based on ns-3 [74].
- **MAREA-core**, the central orchestration layer that manages and coordinates the other two components.

Figure 4.1 shows a high level view of the structure of these modules, which will be described in the following. At the topmost layer, **MAREA-core** leverages a Docker-based architecture [95]. Specifically, **MAREA-CP** and **MAREA-ns3** each run in their own container, ensuring clear separation of simulator-specific requirements and operations. Container lifecycle events, as well as inter-container networking, are managed through Docker Compose. This approach provides a dedicated virtual bridge that enables reliable and lossless TCP-based communication between the two simulators.

Communication between **MAREA-CP** and **MAREA-ns3** is implemented by two logically specular bridge modules, **MAREA-CP-bridge** and **MAREA-ns3-bridge**, each comprised into its respective simulator. These modules handle socket-based data exchange with the Docker network, encapsulating simulator-specific operations while enabling seamless inter-simulator information flow. Full details of such modules, as well as the complete set of features from **MAREA-CP** and **MAREA-ns3**, are later presented in Sections 4.2.1 and 4.2.2 respectively, with now focusing on the **MAREA-CONFIG** logical module.

This module provides a unified repository of YAML configuration files and helper scripts for setting up experiments. At the top level, this module defines:

- Addresses and ports used by Docker Compose to establish a virtual network.
- Shared volumes for data and log passthrough between the host and containers.
- System-wide parameters for launching and coordinating the simulators.
- Shared settings and utilities for configuring experiment-specific parameters (*e.g.* networking stack components), applications installed on each node (*e.g.* autonomous navigation), network topology, and noise specifications on navigation sensors.

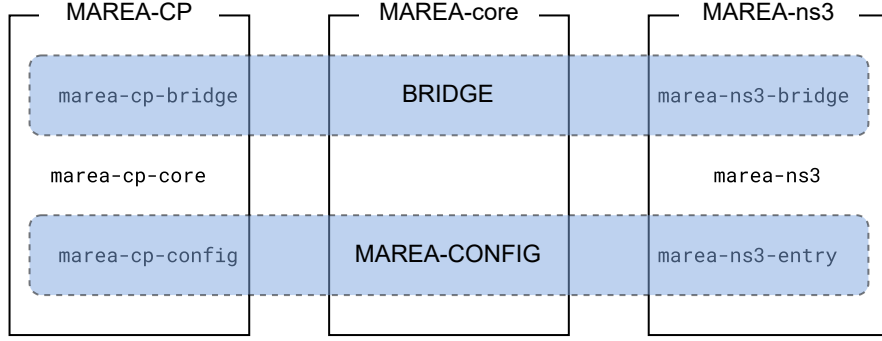


Figure 4.1. Software architecture of MAREA simulator showing high-level perspectives of MAREA-CP and MAREA-ns3, and their internal modules.

Most notably, a one-to-one mapping between topologies of each simulator is employed: each of the N networked agents defined in MAREA-CP is uniquely represented as a network node in MAREA-ns3.

Before delving into the individual descriptions of our proposed extensions, the three kinds of messages exchanged across our joint framework are introduced:

1. **COMM (bi-directional):** Carries application-specific payloads for networked nodes interactions.
2. **GT (MAREA-CP $\rightarrow$ MAREA-ns3):** Conveys ground truth pose and mobility information for networked nodes, as well as the clock needed to maintain a consistent simulation state in both simulators.
3. **BRIDGE_PROT (MAREA-ns3 $\rightarrow$ MAREA-CP):** Contains inter-simulator metadata for aligning the shared synchronization timeline.

A detailed discussion of the synchronization protocol and inter-simulator coordination is later provided in Section 4.2.3.

The overall interaction between components is briefly illustrated in Figure 4.2. Here, dedicated ROS topics are used within MAREA-CP for publishing or retrieving GT or COMM messages (steps 1-2). In MAREA-ns3, apart from SYNC messages (step 5) that are directly generated by Bridge, the rest of messages are retrieved and handled via the ns-3 *TraceSource* mechanism or callbacks (steps 3-4). It is important to note that GT messages are never sent to MAREA-CP (step 3), whereas COMM messages can be exchanged in both inbound and outbound directions (step 4).

The modular design of this simulator caters to several kinds of users, ranging from applications in complex scenarios (*e.g.*, networked UUVs operating in cooperation scenarios) to simpler ones, since the proposed extensions may be also used individually.

4.2.1 MAREA-CP: Extensions to UUV Simulator

As introduced at the beginning of this section, the cyber-physical aspect of the proposed simulation framework is a direct extension of the *UUV Simulator*, leveraging its features for modeling underwater physics, vehicles, and sensors. In what follows, the

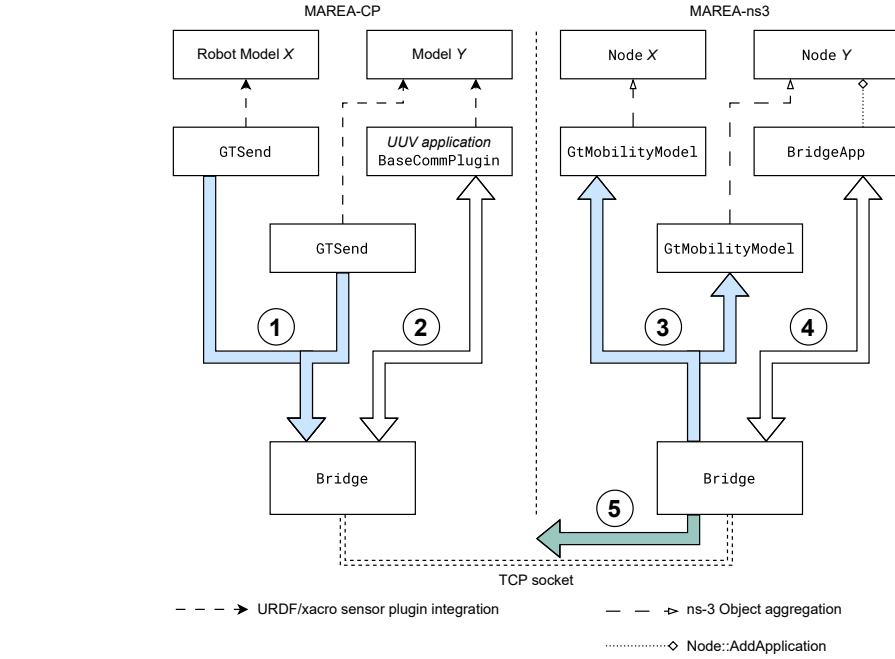


Figure 4.2. Mapping between Robot Model or Model instances in GAZEBO and Node instances in ns-3, as well as exchanges between components of MAREA simulator.

principal logical modules of MAREA-CP, namely MAREA-CP_core, MAREA-CP_bridge, and MAREA-CP_config are outlined.

The MAREA-CP_core comprises several components:

(i.) *The base framework* on which MAREA-CP is built, namely *UUV Simulator*. Together with GAZEBO and ROS integration, it provides a robust set of APIs to model complex underwater scenarios.

(ii.) *A reworked version of the UUV Simulator rextrov2 robot model* that retains the physical structure and properties of the original but is augmented with additional parameters. These parameters allow multiple instances—potentially featuring different equipment or layouts—to be automatically configured and deployed from external configuration files.

(iii.) *Additional custom object models* for roped, fixed, or free-floating buoys that can carry an arbitrary set of sensors (e.g., GPS, acoustic modem, ...). Similar to *rextrov2*, these buoy models are easily configurable from external configuration files.

(iv.) *Custom GAZEBO/ROS model plugins* that extend the framework’s physical entities with configurable APIs supporting communication-related applications in underwater networks (e.g., one/two-way ranging or remote task assignment from smart buoys to mobile nodes). These plugins operate via ROS messages published conditionally or on user demand to topics that the MAREA-CP_bridge subscribes to, and similarly subscribe to MAREA-CP_bridge topics to process messages from the network side.

(v.) *The GT GAZEBO/ROS model plugin*, installable on each physical entity, which generates ground-truth (GT) messages that encode the pose of the associated node at a specified user-defined rate. Crucially, it also supplies the clock to MAREA-ns3

for co-simulation synchronization.

(vi.) *Multiple localization pipelines* that can be configured to work with various subsets of inertial sensors and acoustic ranging. These provide a reliable, plug-and-play approach to localization under most circumstances. These include the approaches presented in Sections 3.2 and 3.3 and a Factor Graph-based pipeline, shown in Appendix A. Additionally, ROS nodes are provided to process sensor measurements for feeding into these localization pipelines. These nodes can also serve as templates for users wishing to implement their own algorithms.

The **MAREA-CP_bridge** offers bridging capabilities between MAREA-CP and MAREA-ns3. This module implements inter-simulator communication on the MAREA-CP side, enabling synchronization between simulations, redirecting ground-truth messages from all networked nodes to MAREA-ns3, and relaying all COMM and BRIDGE_PROT messages to/from MAREA-ns3 over a TCP socket.

Finally, the **MAREA-CP_config** module implements scripts that access the global configuration files provided in MAREA-CONFIG, converting YAML specifications in a single automatically generated ROS `launch` file that is used to launch the simulation on the MAREA-CONFIG side. This is particularly useful for reproducibility concerns, since this file can then be used to launch the same exact simulation any number of times just by using the ROS `roslaunch` utility. Bags and logs produced during experiments are then automatically stored on the host machine (along a summary of the whole experiment parameters) for users' own elaboration and/or evaluation scripts.

4.2.2 MAREA-ns3: Extensions to ns-3

Three ns-3 modules are proposed in this Section, namely **marea-ns3**, **marea-ns3-bridge**, and **marea-ns3-entry**. The first module is dedicated to implementing novel features in underwater network simulation in ns-3, whereas the other two modules enable the exploitation of realistic physics from GAZEBO in ns-3.

marea-ns3 module provides a framework for implementing auto-configurable multimodal and multi-stack networking solutions, with common performance metrics already implemented. This module is natively compatible with UAN Framework and other external modules based on such ns-3 API can be integrated with minimal efforts, most notably WOSS [96] for Bellhop-based acoustic propagation. The most important features provided by this module are described next.

(i.) *Unified API for entire networking stack.* General and robust APIs for implementing custom models at PHY, MAC, and NET layers in the protocol stack, with cross-layering communication based on a topic-based publisher/subscriber model, namely **CrossLayer**. Advanced users may easily leverage such APIs for their own implementation of novel protocols. Moreover the designed structure retains compatibility with ns-3 Socket API, allowing the usage of pre-existing ns-3 Applications (*e.g.* traffic generation models) directly in the underwater domain.

(ii.) *Multimodal communication capabilities.* Several channel and PHY layer models for different underwater communication technologies (acoustic, wireless optical, electromagnetic) are provided via a technology-agnostic API. Available upper-layer protocols may be used with any kind of PHY layer or combinations of them.

(iii.) *Support for multiple and self-configuring networking stacks.* Adaptive stacks consisting of multiple PHY and MAC layers, arranged in one or more **Devices** (**DEV**), in turn handled by **NET** layers allow for a thorough definition of protocol stack on each node. As briefly depicted in Figure 4.3, this level of detail in protocol stacks is transparently made available through the ns-3 Socket API. At the same time, advanced users may leverage a rich API for implementing specific use cases that require deeper protocol stack control. For example, **Socket** and **Device** instances respectively hold their own packet queues, and **NET/MAC** instances access them for processing according to protocols' logic. Throughout the simulation, layer-specific **Configurator** instances can be set by users to monitor performance metrics and apply adjustments to parameters or dynamically swap to other protocols in order to optimize the entire stack.

(iv.) *Implementations of common networking protocols.* Pure ALOHA, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with truncated BEB, and Time Division Multiple Access (TDMA) are provided as MAC layers. At the **NET** layer, routing by flooding (with randomized delay and/or bounded Time-to-Live options) and static routing with a fine-grained routing table (next-hop address and outgoing **Device** based on destination, previous hop, and incoming **Device** for a packet). These protocols can also serve as templates for users interested in implementing their own solutions.

(v.) *Network performance metrics.* Available metrics for MAC/**NET** layers: queue latency, queue status, queue drop ratio, protocol overhead ratio, throughput, and transmission attempts. **NET**-only metrics: routing path, which doubles as hopcount. Finally, packet delivery ratio and end-to-end latency at the application level, and current miscellaneous metrics only consist in energy consumption. This is an effort in improving reproducibility and comparisons, since metrics are implemented uniformly and independently of protocol stack details.

marea-ns3-bridge module provides bridging capabilities between ns-3 and MAREA-CP, therefore enabling realistic underwater physics and navigation in network simulations. This module implements inter-simulator communication on the ns-3 part, allows synchronization between simulations, and handles mobility and communication needs of networked robots. A dedicated **MobilityModel**-derived class (*i.e.* **GtMobilityModel**) is provided, and regular updates received over a TCP socket [97] connection from MAREA-CP are used to appropriately control synchronization and robots' positions (more details in Section 4.2.3).

A proxy ns-3 Application is also provided, namely **BridgeApp**, that serves as entry and exit point of the networking stack: incoming **COMM** messages from the bridge are in fact captured by this application on a source node and inserted in its networking stack for processing; instead, a receiving node will write an outbound **COMM** message through the bridge once a payload packet is completely processed bottom-up within its networking stack. An additional feature provided by this application is tracking delays incurred at each layer during the protocol stack traversal, *e.g.* back-off time at **NET** layer, awaiting an assigned transmission slot at **MAC** layer, etc. Delays are tracked w.r.t. a source node and inserted in the **COMM** message as metadata for receivers to exploit; time-sensitive applications might then use this information to

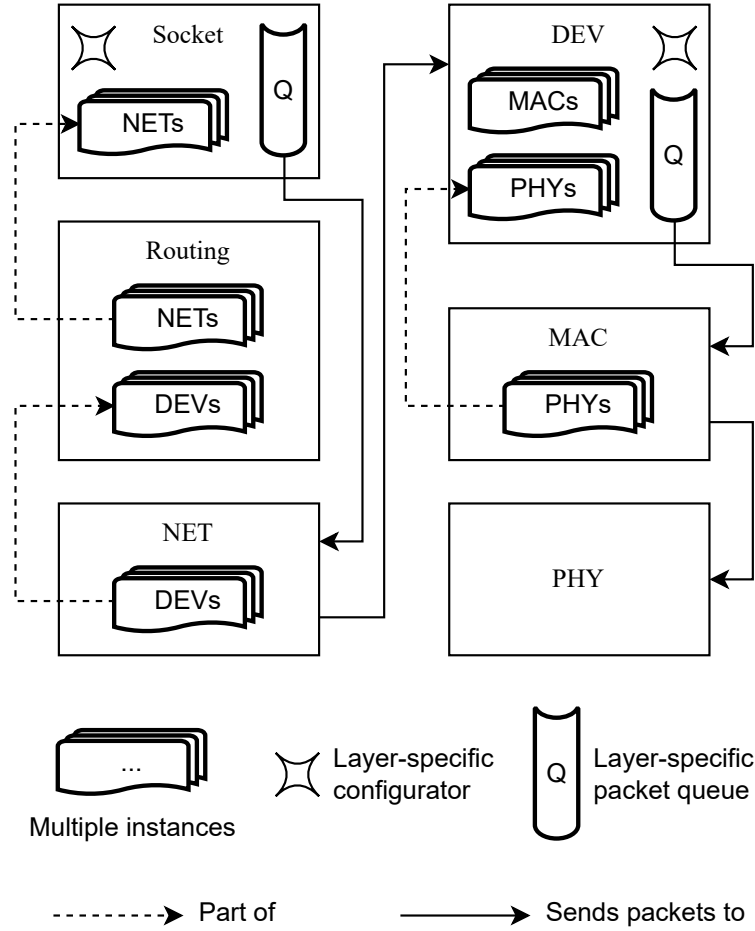


Figure 4.3. Support for multiple networking stacks in **MAREA-ns3**; one or more NET, MAC, or PHY instances can compose a node’s networking stack, transparently used through the ns-3 Socket API, and enabled for self-configuration via layer-specific configurators.

compensate for stack delays¹.

marea-ns3-entry provides a straightforward dependency to combine the two previously described ns-3 modules, alongside a user-friendly helper API for configuring experiments via YAML files.

4.2.3 Inter-simulator communication and synchronization protocol

Extensions proposed in previous sections are able to communicate via TCP sockets using a minimal yet expressive protocol, sufficiently flexible for many applications.

Communication. The inter-framework communication protocol employs a line-based protocol in which string messages are exchanged through TCP sockets, where application-specific data is distinguished by respective prefixes. Figure 4.4 depicts an example of communication between simulators, that is described in the following. **MAREA-CP** applications leveraging remote communication will create their payload

¹This is equivalent to the case where a message has awaited in a buffer; upon reaching the front

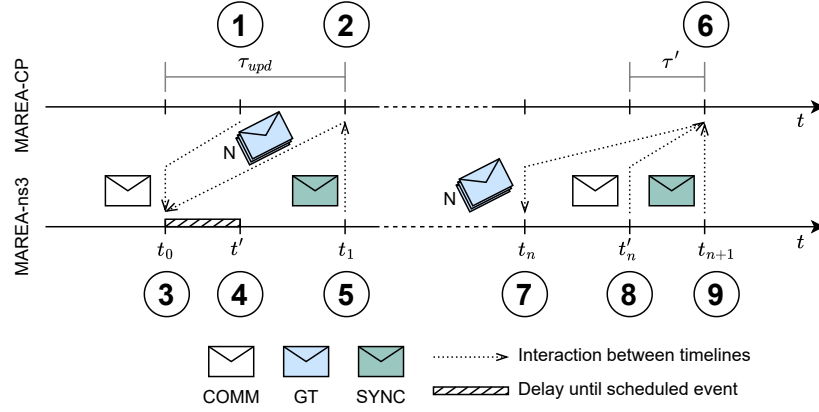


Figure 4.4. Inter-simulator communication protocol through sockets.

message at an arbitrary time t' (step 1); this message is then sent to MAREA-ns3 as a COMM message, scheduling it for processing at the desired timestamp t' (step 4). In order to actually process it, the next GT-based update from MAREA-CP is needed: upon reaching timestamp t_1 , N instances of GT messages are created (step 2) which will cause MAREA-ns3 to advance its timeline from t_0 to t_1 . Any scheduled networking event occurring during this interval is therefore executed. Analogously, in case a COMM message is produced by MAREA-ns3 during this interval, its actual notification to MAREA-CP will occur at time t_1 . For example, in Figure 4.4 at step 8, a payload message is created at timestamp t'_n in MAREA-ns3, however it will be conveyed when MAREA-CP has already reached timestamp t_{n+1} , delayed by τ' . τ' is referred to as *desynchronization delay*: this is due to the design differences between the two simulators, and similarly tracked as protocol stack delays. The following paragraph describes how the proposed approach tackles the synchronization delay.

Synchronization. GAZEBO and ns-3 are structurally different simulators: the former presents a continuous time progression, with physics being evaluated according to differential equations modeling the evolution of internal states, whereas the latter is based on discrete events. To overcome this mismatch, a synchronization protocol to ensure timelines between simulators are properly aligned is devised.

Due to the typically faster progression in event-based simulators, the first part of the proposed synchronization mechanism only affects MAREA-ns3: by using blocking socket reads, and given the single-thread single-process execution of ns-3, this approach easily prevents the timeline in ns-3 from ever advancing ahead UUV's. The socket read returns whenever MAREA-CP sends any message through, with the timeline in ns-3 only advancing in case of receiving a GT message with a timestamp greater than current ns-3 timestamp. Recalling Figure 4.4, MAREA-ns3 is at timestamp t_0 and is blocked on a socket read; a COMM message is received with timestamp t' , but the timeline does not advance and remains at t_0 (step 3). A further socket read returns the first GT message having timestamp equal to t_1 : at this point, mobility information is updated and the next blocking socket read will be performed once MAREA-ns3 reaches timestamp t_1 . This results in any event scheduled to be executed,

of the buffer, it would be apposed a new timestamp right before transmission.

such as the payload message entering the networking stack at timestamp t' (step 4).

The second part of the synchronization protocol is devoted to preventing MAREA-CP from advancing too far ahead of the event-driven simulation. After sending N GT messages, MAREA-CP pauses physics updates in GAZEBO until a SYNC message is received from MAREA-ns3. The SYNC message is sent once MAREA-ns3 has read the totality of N GT messages from the socket, being N known as per configuration. This mechanism is highlighted in Figure 4.4: at step 2, MAREA-CP sends a GT message for each robot, pausing GAZEBO updates and freezing its timeline at t_1 . After processing N mobility messages, MAREA-ns3 advances to timestamp t_1 and sends the SYNC message (step 5). With this reception, MAREA-CP resumes GAZEBO updates and is thus able to advance its timeline as it normally would. This simple mechanism yields a desynchronization delay τ' between simulations that is upper-bounded by the GT message transmission period τ_{upd} . Such delay is mostly noticeable between steps 8 and 9 (Figure 4.4), since MAREA-ns3 is only able to notify the reception of any COMM message after it is allowed to proceed in its timeline. It is remarked that, since τ_{upd} is configurable, the granularity of the τ' delay can be tuned to fit scenarios with different dynamics. In Section 4.3, a brief analysis of the implications stemming from choices in τ_{upd} value is provided.

4.2.4 Modularity and typical usage

The modular design of the simulator caters to a wide variety of use cases, without necessarily requiring both extensions at once. For example, when only developing networking solutions tailored for static sensor networks, `marea-ns3` alone might be sufficient in ns-3 modules or simulation scripts. Conversely, use cases that do not require a framework as extensive as the one provided above, yet needing a networking stack based on ns-3 in their robotics experiments, may only leverage `marea-ns3-bridge` in simulation scripts.

4.3 Use Case Validation Example

In this section a use case validating the motivations behind the development of the MAREA framework is provided, focusing on performance implications and tradeoffs depending on the chosen simulation parameters.

The application considered in this benchmark is a 1-Way LBL ranging towards N_a AUVs and initiated by N_b moored buoys. This kind of setting, despite apparently simple, highlights the critical aspects affecting the integration of the two simulators, i.e., communication timeliness and timelines synchronization. In fact, the accuracy of the range computation heavily relies on the robot being able to compute a distance estimate from the 1-Way Travel Time of the acoustic signal, net of any time delay possibly introduced by simulation timeline misalignment between MAREA-CP and MAREA-ns3. In the proposed setting, AUVs follow a pre-determined trajectory at an average speed of ~ 1.5 m/s and localize themselves by means of inertial navigation and range estimates. For their networking stacks, AUVs and buoys are equipped with acoustic modems, employing a TDMA MAC and static routing (unicast) in their protocol layers. TDMA configuration is such that 1 slot is assigned to AUVs

Parameter	Symbol	Value(s)
Number of AUVs	N_a	1, 5, 10, 20
Number of buoys	N_b	3
GT update frequency	τ_{upd}	1, 2, 10 Hz

Table 4.1. Simulation parameters.

and 2 consecutive slots to each buoy, with slots having equal duration (0.5s). Table 4.1 summarizes the parameters explored in the simulations for this experiment.

Hereby the thesis focuses on the granularity of inter-simulation synchronization (τ_{upd}), analyzing how it affects overall simulation duration (T_{Wall}), desynchronization between timelines (τ' , see Sec. 4.2.3), and the accuracy of distance computation from the acoustic ranging through the ranging error ϵ_{range} . ϵ_{range} is defined as the absolute difference between the estimated range and the ground-truth distance, computed when a ranging message is received by an AUV at time $t + k\tau_{upd} + \tau'$. This metric is extremely sensitive to position information from the points of view of GAZEBO and ns-3: as seen in Figure 4.5, whenever a ranging message is received, the position recorded in ns-3 might not necessarily be up-to-date, as it could possibly refer to an earlier GT update round of GAZEBO's ground truth, at time $t + k\tau_{upd}$. This is due to AUVs' dynamics: in fact, as mentioned in Sec. 4.2.3, the two simulators are synchronized net of τ' . Thus, the dynamics of physical entities in experiments proportionally amplifies the impact of τ' on the inter-simulator synchronization.

Since τ' directly depends on τ_{upd} , the choice of the latter will be driven by the nature of the experiment. Figure 4.6 and Figure 4.7 depict the impact of

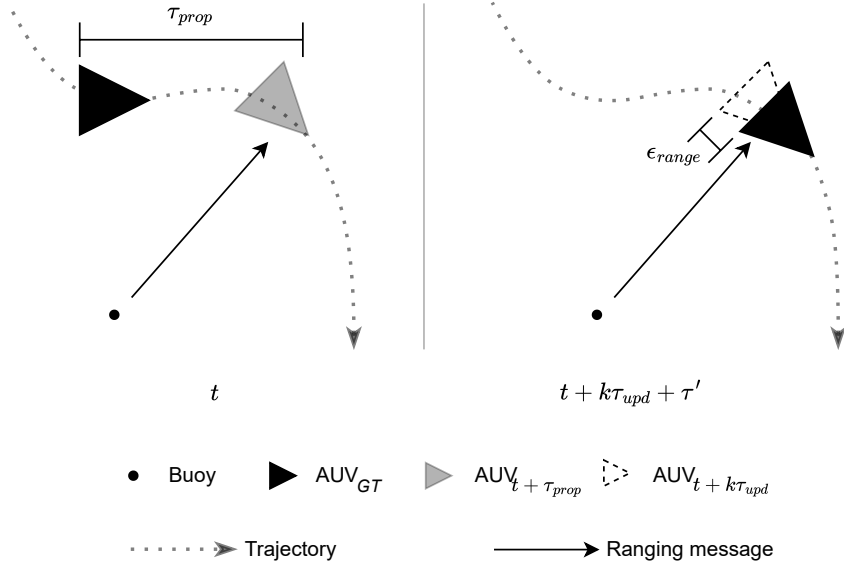


Figure 4.5. Explanation of the ϵ_{range} metric in the ranging use case. τ_{prop} is the propagation delay for a communication initiating at time t ; w.l.o.g., the communication is received after k GT updates plus desynchronization delay τ' .

the GT update frequency τ_{upd} choice on ranging accuracy and simulation runtime, respectively. The scenario having $N_a=1$ and $\tau_{upd}=0.1s$ (10 Hz) as the baseline is considered. At an AUV speed of $\sim 1.5 m/s$, the distance traveled during a $\tau'=0.09s$ is approximately 0.14m; this aligns with the observed average ranging error of $\sim 0.15m$ throughout the baseline experiment, with a simulation duration T_{Wall} being 32% greater than T_{ROS} ², namely the simulated timeline. For completeness, τ_{upd} setting at 0.01s has also been explored, but no increase in ranging accuracy in simulations was found, at the cost of an exponentially increasing overhead.

Decreasing the GT update frequency improves on the synchronization overhead, thus yielding durations closer to a GAZEBO-only setting, at the cost of a slightly less precise ranging. In fact, only a 0.09m and 0.18m increase in average ranging errors is observed with respect to the baseline with τ_{upd} set to 0.5s (2 Hz) and 1s (1 Hz) respectively. These τ_{upd} settings, however, allow a reduction in T_{Wall} overhead to just 8% and 3% more than T_{ROS} , making the simulation practically real-time.

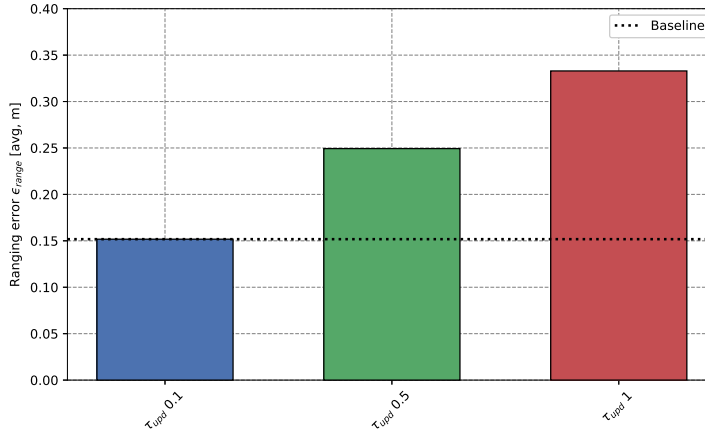


Figure 4.6. Average ranging error, varying GT update frequency τ_{upd} .

Shifting the focus on scalability of the presented framework, experiments are conducted by increasing N_a . This results in more traffic between simulators across the virtual network, since both the number of GT updates and COMM messages exchanged is higher. As seen in Figure 4.7, MAREA is able to sustain this increased load with T_{Wall} value increases in ranges 3-9% (1 Hz), 8-45% (2 Hz) and 28-60% (10 Hz), therefore without drastically affecting the simulation duration. Finally, Figure 4.8 shows the desynchronization delay τ' across different settings. As mentioned in Sec. 4.2.3, the synchronization protocol enforces τ_{upd} to be the upper bound for τ' . Across the experiments and regardless of N_a settings, values for τ' equal to 0.09s, 0.46s, and 0.96s are observed during scenarios performing synchronization every 0.1s, 0.5s, and 1s respectively.

²In order to achieve a high-fidelity physics simulation, this experiment's settings in GAZEBO are such that any simulation executes no faster than real time. As a result, even without the overhead introduced by MAREA-ns3, the best case of T_{Wall} overhead w.r.t. T_{ROS} is 0%, *i.e.* the two values being equal.

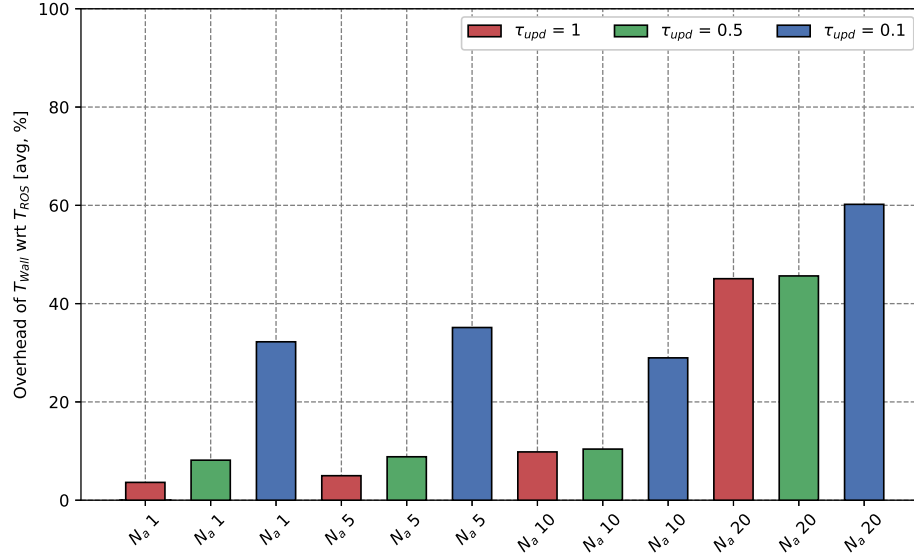


Figure 4.7. Average T_{Wall} overhead w.r.t. T_{ROS} , varying number N_a of AUVs and GT update frequency τ_{upd} .

4.4 Conclusions

This Section introduced **MAREA**, the first co-simulation framework for marine multi-robot experiments addressing both accurate physics models and a comprehensive multimodal networking protocol stack. **MAREA** is distributed as a containerized solution to increase ease of adoption, with physical modeling provided via GAZEBO whereas ns-3 is tasked with network simulation.

The proposal consists of several extensions to the above simulators s.t. a wide range of simulation-based experiments in the underwater domain can be performed with low effort yet retaining a high degree of flexibility and realism in the results. Particular emphasis is placed on easily enabling the study of multi-robot scenarios encompassing current trends, such as adaptive networking stacks and multimodal communication capabilities beyond acoustics, *e.g.* wireless optical and electromagnetic solutions.

Finally, an analysis confirming the suitability of our simple approach to simulators' synchronization is provided, highlighting the trade-off between a low-overhead simulation and a more accurate one. Furthermore, the scalability of **MAREA** to moderately large multi-robot simulations is shown, while keeping tight synchronization between GAZEBO and ns-3 simulators, demonstrating its fitness in facing novel challenges in the field.

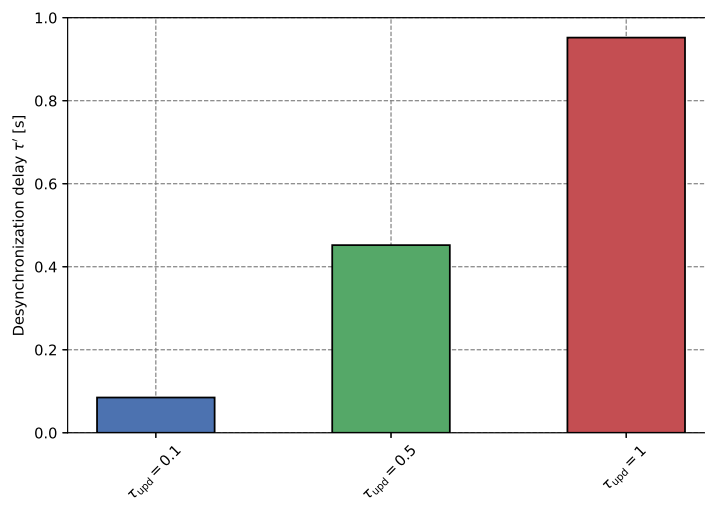


Figure 4.8. Desynchronization delay τ' , varying GT update frequency τ_{upd} .

Chapter 5

Conclusions and Future Works

5.1 Summary of Contributions

This thesis investigated the challenges to modern and trustworthy under-sea autonomy, focusing on acoustic-based localization and high-fidelity simulation. Two of the most important pillars of intelligent underwater systems and applications, namely noise-aware navigation and realistic cyber-physical networked simulation, have been addressed through a combination of field-tested estimation algorithms and the novel MAREA co-simulation framework.

First, a deterministic acoustic localization system was developed, capable of achieving meter-level localization accuracy even with modest hardware and no prior environmental calibration. However, experimental analysis revealed the limitations of fixed-noise models in the face of non-stationary underwater dynamics and sensor noise. To address this, the thesis introduced an adaptive localization architecture based on an Extended Kalman Filter with environmental and sensors parameters estimation. This approach compensates for time-varying disturbances and sensor bias online, achieving over 50 % improvement in localization accuracy compared to static models in both synthetic and real-world conditions. This result demonstrates that a robust, resilient underwater navigation system does not only require sophisticated hardware, but also intelligent model adaptation.

Complementing this, the thesis presented MAREA, the first cyber-physical simulator combining physics-based vehicle dynamics with acoustic communication and signal propagation modeling. By tightly integrating a modified ns-3 stack with a physics-aware simulator, MAREA supports realistic, multimodal underwater experimentation. Its ability to reproduce field-observed phenomena such as desynchronization, communication loss, and propagation delay confirms its fidelity and applicability for both academic and industrial use cases.

Together, these contributions form a robust, adaptable pipeline from simulation to real deployment, facilitating repeatable, low-cost experimentation and robust navigation in unpredictable underwater environments.

5.2 Future Directions

Looking forward, the future of underwater robotics lies in fully integrated, autonomous ocean systems that blend learning, simulation, and real-time control.

The MAREA framework sets the stage for the development of **Digital Twin** architectures, virtual models that maintain a synchronized representation of the physical system and the environment. Integrating online AUV telemetries and incorporating live oceanographic data with the simulated twin in a live data fusion fashion, could allow real time drift correction and predictive modeling. The digital twin can also be employed as an anticipatory planning module that simulates near-future outcomes and informs the on board decision modules in uncertain environments, enhancing vehicles situation awareness in data-informed systems.

Reducing the “reality gap” between simulation and field deployment is a critical task. The MAREA framework paves the road to enhancing the **Sim-to-Real** paradigm in underwater scenarios. Using MAREA-generated data and Domain Randomization to train surrogate models (e.g., neural networks) has the potential to approximate unmodeled dynamics and set informed priors to automatically initialize filter parameters or sensor alignment during the missions.

5.3 Final remarks

This thesis demonstrates how noise aware probabilistic methods, adaptive estimation, and realistic cyber-physical modeling can converge to push the boundaries of underwater autonomy. By pairing robust algorithms with realistic simulations, it becomes possible to test, deploy, and iterate faster and more safely, ultimately unlocking broader and more impactful use of autonomous underwater systems. As the oceans become more accessible to robotic systems, the importance of reliable, intelligent, and explainable autonomy will grow.

This thesis offers a step toward this vision, where real and virtual systems work side by side to explore, understand, and protect the last frontier on Earth.

Publication List

1. **Luca Iezzi**, Emanuele Giona, and Chiara Petrioli. MAREA: a Multimodal Co-Simulation Framework for Marine Networked Agents. [*not peer reviewed yet*].
2. **Luca Iezzi**, Chiara Petrioli, and Stefano Basagni. An Adaptive Extended Kalman Filter for State and Parameter Estimation in AUV Localization. In ICC 2023 - IEEE International Conference on Communications, pages 3932–3938, 2023.
3. Irene Tallini*, **Luca Iezzi***, Petrika Gjanci, Chiara Petrioli, and Stefano Basagni. Localizing Autonomous Underwater Vehicles: Experimental Evaluation of a Long Baseline Method. In 2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS), pages 443–450, 2021.

*Equal contribution.

Best Paper Award Runner-up.

Appendix A

A Practical MAREA Example on a Factor Graph Localization Pipeline

This Appendix showcases the MAREA framework capabilities on a realistic application example, where multiple AUVs navigate and localize themselves by means of inertial navigation and acoustic ranging with external acoustic beacons. The localization machinery employed by the vehicles to fuse the sensor measurements and estimate their state during navigation is a Factor Graph iSAM2-based pipeline, which is included in MAREA. Such method has only been validated in simulation, through the MAREA framework itself. As such, it has not yet been tested in real field experiments and does not yet provide a detailed analysis on accuracy and stability for a proper State-of-The-Art comparison and publication. Nonetheless, as it provides the reader with some insights on the MAREA simulator, it is presented in the following.

After showing the system model and the iSAM2 formulation adopted in the state estimation pipeline, this Appendix proceeds in describing the simulated scenario and analyzing the performance of the localization approach.

A.1 System Model

State vector

At each timestep $k \in \{0, \dots, K\}$, the 15-parameter navigation state is

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{p}_k \\ \mathbf{q}_k \\ \mathbf{v}_k \\ \mathbf{b}_k^a \\ \mathbf{b}_k^g \end{bmatrix}, \quad \mathbf{p}_k \in \mathbb{R}^3, \mathbf{q}_k \in \text{SO}(3), \mathbf{v}_k, \mathbf{b}_k^a, \mathbf{b}_k^g \in \mathbb{R}^3.$$

where the quaternion is expressed in the tangent space $\mathfrak{so}(3)$. Similarly to the localization approaches shown in Sections 3.2 and 3.3, N beacons with positions $\mathbf{p}_{b,i} \in \mathbb{R}^3$ yield asynchronous slant-ranges.

Sliding-window (fixed-lag) posterior

To bound memory and computation, optimization is confined to the most recent n epochs,

$$\mathcal{W}_k = \{k - n + 1, \dots, k\}, \quad n \text{ fixed.}$$

Variables with indices $< k - n + 1$ are marginalized exactly once they leave the window, producing a summary factor ϕ_{k-n}^{marg} that freezes their linearized information. With $\mathcal{R}$ and $\mathcal{D}$ the set of range and depth timestamps, respectively, the posterior at epoch k factorizes as

$$p(\mathbf{x}_{\mathcal{W}_k} \mid \mathcal{Z}_{0:k}) \propto \underbrace{\phi_{k-n}^{\text{marg}}}_{\text{fixed}} \phi^{\text{prior}} \prod_{j=k-n+1}^{k-1} \phi_j^{\text{imu}} \prod_{(j,i) \in \mathcal{R} \cap \mathcal{W}_k} \phi_{j,i}^{\text{rng}} \prod_{j \in \mathcal{D} \cap \mathcal{W}_k} \phi_j^{\text{dep}}.$$

MAP optimisation over the window

The windowed estimate solves

$$\mathbf{x}_{\mathcal{W}_k}^* = \underset{\mathbf{x}_{\mathcal{W}_k}}{\text{argmin}} \left(\|r_{k-n}^{\text{marg}}\|^2 + \|r^{\text{prior}}\|^2 + \sum_{j=k-n+1}^{k-1} \|r_j^{\text{imu}}\|_{\Sigma_{\text{imu}}^{-1}}^2 + \sum_{(j,i) \in \mathcal{R} \cap \mathcal{W}_k} \|r_{j,i}^{\text{rng}}\|^2 + \sum_{j \in \mathcal{D} \cap \mathcal{W}_k} \|r_j^{\text{dep}}\|^2 \right), \quad (3)$$

where the dimension of $\mathbf{x}_{\mathcal{W}_k}$ is bounded by dn with d the length of the sliding window.

iSAM2 in fixed-lag mode

Bayes-tree formulation. Linearising Equation (3) yields $\mathbf{A}\delta\mathbf{x} = \mathbf{b}$ with sparse $\mathbf{A}$. The variable elimination factorizes the Gaussian $\exp(-\frac{1}{2}\|\mathbf{A}\delta\mathbf{x} - \mathbf{b}\|^2)$ into a Bayes tree whose cliques store conditional densities, as seen in 2.3.3. Only the cliques who have variables in $\mathcal{W}_k$ are kept in memory, and ϕ_{k-n}^{marg} sits at the root.

Incremental update. Upon reception of a new batch of factors (IMU span, range, depth), iSAM2 performs (i) *partial relinearization* for variables whose linearization error exceeds $\eta_{\text{rel}} = 0.1$, (ii) *local re-elimination* restricted to the involved cliques, and (iii) one back-substitution pass through the tree to recover $\delta\mathbf{x}$.

Residuals and Jacobians

The factor types, and their respective Jacobians $\mathbf{J}$, used in the present formulation are presented in the following.

Prior factor The initial belief is encoded by the couple factor-Jacobian

$$r^{\text{prior}} = \Sigma_0^{-\frac{1}{2}} (\mathbf{x}_{k-n+1} - \bar{\mathbf{x}}), \quad \mathbf{J}^{\text{prior}} = \Sigma_0^{-\frac{1}{2}}.$$

IMU pre-integration factor Between between i and $j = i+1$, let $\tilde{\alpha}, \tilde{\beta}, \tilde{\mathbf{R}}$ denote the pre-integrated displacement, velocity and rotation computed with bias $\bar{\mathbf{b}}_i$ over Δt , $\mathbf{R}_i = \mathcal{R}(\mathbf{q}_i)$, and g the gravity vector. With $\delta \mathbf{b}_i^a = \mathbf{b}_i^a - \bar{\mathbf{b}}_i^a$, $\delta \mathbf{b}_i^g = \mathbf{b}_i^g - \bar{\mathbf{b}}_i^g$ and bias Jacobians $\mathbf{J}_\alpha^{(\cdot)}, \mathbf{J}_\beta^{(\cdot)}, \mathbf{J}_\theta^g$ returned by the pre-integrator, the IMU pre-integration factor is:

$$r_{ij}^{\text{imu}} = \begin{bmatrix} \mathbf{R}_i^\top (\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i \Delta t - \frac{1}{2} g \Delta t^2) - \tilde{\alpha} - \mathbf{J}_\alpha^a \delta \mathbf{b}_i^a - \mathbf{J}_\alpha^g \delta \mathbf{b}_i^g \\ \mathbf{R}_i^\top (\mathbf{v}_j - \mathbf{v}_i - g \Delta t) - \tilde{\beta} - \mathbf{J}_\beta^a \delta \mathbf{b}_i^a - \mathbf{J}_\beta^g \delta \mathbf{b}_i^g \\ \text{Log}(\tilde{\mathbf{R}} (\mathbf{R}_i^\top \mathbf{R}_j)) - \mathbf{J}_\theta^g \delta \mathbf{b}_i^g \\ \mathbf{b}_j^a - \mathbf{b}_i^a \\ \mathbf{b}_j^g - \mathbf{b}_i^g \end{bmatrix},$$

with covariance Σ_{imu} from the integrator.

Let $\mathbf{a} = r_\alpha^{\text{imu}}$, $\mathbf{v} = r_\beta^{\text{imu}}$, $\boldsymbol{\theta} = r_\theta^{\text{imu}}$, $\mathbf{A} = \mathbf{J}_r^{-1}(\boldsymbol{\theta})$ and $[\cdot]$ the skew operator. With respect to the error states $\delta \mathbf{x}_i, \delta \mathbf{x}_j \in \mathbf{R}^{15}$ the Jacobian of the IMU factor is

$$\mathbf{J}_{ij}^{\text{imu}} = \begin{bmatrix} -\mathbf{I} & -[\mathbf{a}] & -\mathbf{I}\Delta t & -\mathbf{J}_\alpha^a & -\mathbf{J}_\alpha^g & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -[\mathbf{v}] & -\mathbf{I} & -\mathbf{J}_\beta^a & -\mathbf{J}_\beta^g & \mathbf{0} & \mathbf{0} & \mathbf{R}_i^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{A} & \mathbf{0} & \mathbf{0} & -\mathbf{J}_\theta^g & \mathbf{0} & \mathbf{A} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

Acoustic range factor Let the predicted range be $h(\mathbf{x}_k) = \|\mathbf{p}_k - \mathbf{p}_{b,i}\|$ and measurement $d_{k,i}$. Then:

$$r_{k,i}^{\text{rng}} = \frac{h(\mathbf{x}_k) - d_{k,i}}{\sigma_{r,i}}, \quad \mathbf{J}_{k,i}^{\text{rng}} = \frac{1}{\sigma_{r,i}} \begin{bmatrix} \hat{\mathbf{n}}_{k,i}^\top & \mathbf{0}_{1 \times 12} \end{bmatrix}, \quad \hat{\mathbf{n}}_{k,i} = \frac{\mathbf{p}_k - \mathbf{p}_{b,i}}{h(\mathbf{x}_k)}.$$

Depth factor For a depth measurement z_k^{dep} ,

$$r_k^{\text{dep}} = \frac{e_3^\top \mathbf{p}_k - z_k^{\text{dep}}}{\sigma_z}, \quad \mathbf{J}_k^{\text{dep}} = \frac{1}{\sigma_z} \begin{bmatrix} e_3^\top & \mathbf{0}_{1 \times 12} \end{bmatrix}, \quad e_3 = [0 \ 0 \ 1]^\top.$$

A.2 Simulation Experiments

The simulation involves four AUVs, each following a pre-determined route, and three acoustic beacons (see Figure A.2), deployed in an area of $\sim 250 \times 150\text{m}$ and a maximum depth of 70 m.

During the route, each AUV navigates at an average speed of $\sim 1.5 \text{ m/s}$ and localizes itself by means of inertial navigation and acoustic ranging with the deployed beacons, according to the Factor Graph model presented in the previous Section, with a window size $d = 20$.

AUV and buoys are equipped with acoustic modems, employing a TDMA MAC and static routing (unicast) in their protocol layers. TDMA configuration is such that 1 slot is assigned to AUVs (in which the vehicle broadcasts the ping to initiate ranging) and 2 consecutive slots to each buoy, with slots having equal duration (0.5s)

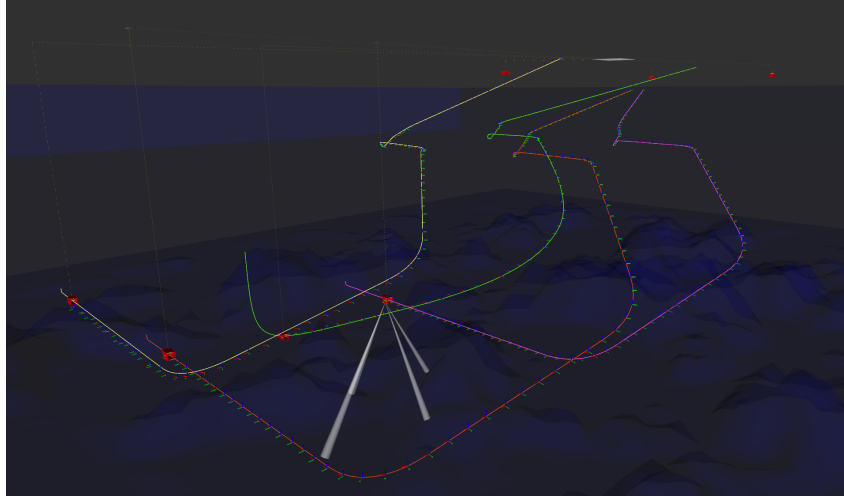


Figure A.1. Simulated environment of the proposed scenario. Four AUVs (with corresponding four differently colored trajectories and real time position and orientation estimates) in a scenario with four deployed beacons (red boxes in the background).

The $\mathbf{GT}$ update frequency of the simulator τ_{upd} is set to 2 Hz. The IMU carried by the AUV has been modeled to behave according the noise characteristics of a medium-grade commercial sensor. Each buoy is modeled as a physical object that is subject to the flow of the surrounding water mass, including water currents. In this experiment, each beacon is suspended at about 1.5 m deep using the said buoys on the top end, and a rope anchored to the seabed. The rope has been modeled as a GAZEBO plugin according to the following model. Let

- $\mathbf{p}_a \in \mathbf{R}^3$ be the *anchor point*;
- $R_{\max} > 0$ the maximum rope length;
- $\mathbf{p}_\ell(t)$ the world frame position of the (buoy) link ℓ at simulation time t ;
- $\hat{\mathbf{d}}_\ell = \frac{\mathbf{p}_a - \mathbf{p}_\ell}{\|\mathbf{p}_a - \mathbf{p}_\ell\|}$ the unit vector pointing from the link to the anchor.

At every GAZEBO update step the plugin computes the distance

$$d_\ell(t) = \|\mathbf{p}_\ell(t) - \mathbf{p}_a\|.$$

If $d_\ell(t) \leq R_{\max}$ no action is taken; otherwise a restoring force

$$\mathbf{f}_\ell(t) = k(d_\ell(t) - R_{\max}) \hat{\mathbf{d}}_\ell \quad (k = 10^6 \text{ N/m})$$

is injected into the physics engine. The gain k plays the role of an (isotropic) rope stiffness and can be tuned.

The environmental-dependent nuisances that affect the sound propagation (see Section 2.1) are modeled through BELLHOP (see Section 4.2.2). This kind of accurate modeling of (i) the beacons and AUVs motion, (ii) the underwater sound propagation and (iii) the networking stack delays, make it so the noise on the range measures

computed by the AUV is not injected as random noise by the user, but is dependent on the actual physical parameters being considered, making the simulation more realistic.

Table A.1 summarizes the main parameters used in the described simulated scenario.

Quantity	Sym.	Unit	Value
Window length	d	-	0.4
Gyro process variance	σ_{ω}^2	rad^2/s^2	0.01
Accel. process variance	σ_a^2	m^2/s^2	0.01
Gyro-bias variance	$\sigma_{b_g}^2$	rad^2	0.01
Accel.bias variance	$\sigma_{b_a}^2$	m^2/s^4	0.01
Beacon-pos. variance	σ_b^2	m^2	0.5
Range-meas. variance	σ_r^2	m^2	0.5

Table A.1. System and noise parameters used in the simulated scenario.

The accuracy of the presented approach is evaluated in terms of 3D localization error (RMSE). Figure A.2 shows the estimated trajectory versus the ground truth. The RMSE obtained by the proposed method relative to each AUV is centimeter-to-

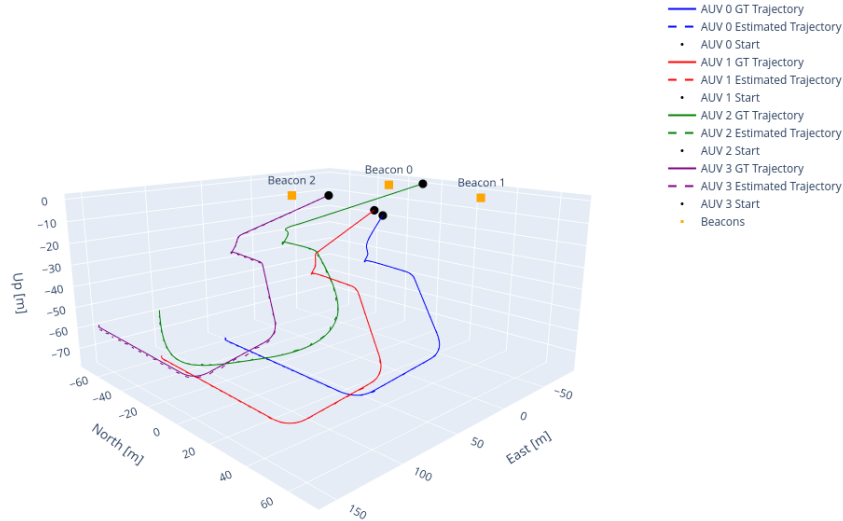


Figure A.2. Four AUVs' Ground Truth (solid trajectories) versus the estimate given by the proposed approach (dashed trajectories).

decimeter level (0.14 m, 0.1 m, 0.33 m and 0.45 m, respectively for each vehicle), and is shown in Figure A.3.

However, as discussed in the opening of this Appendix, a thorough error characterization and in depth analysis of the performance metrics of the presented approach is yet to be carried out, and is deferred to future work.

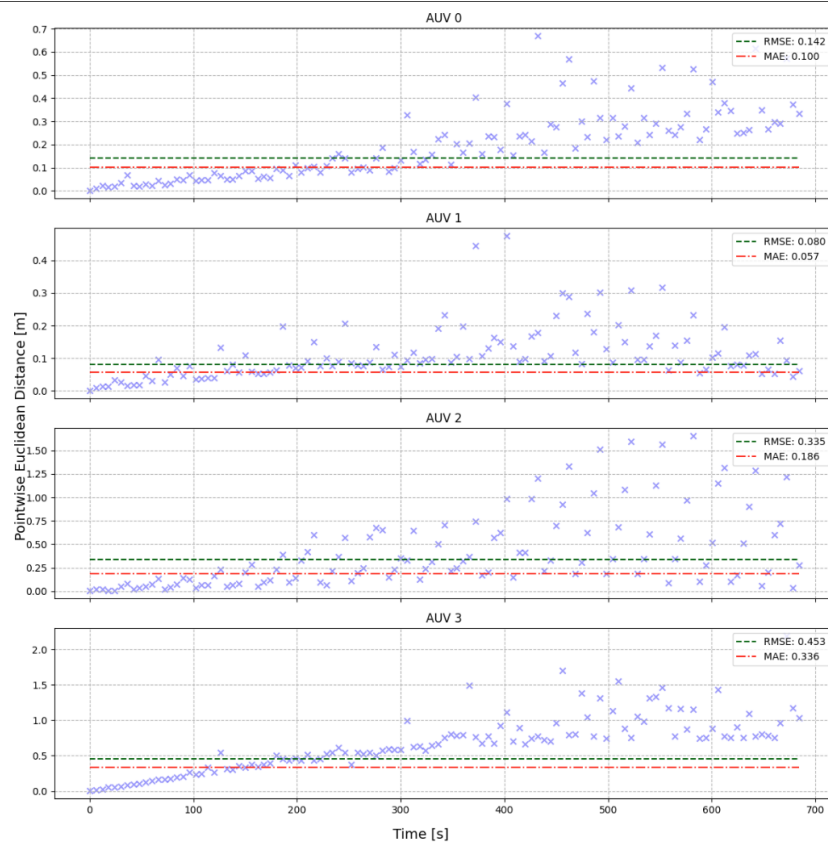


Figure A.3. Localization error (RMSE, MAE) of the proposed approach for each deployed AUV.

Bibliography

- [1] Ian Akyildiz, Dario Pompili, and Tommaso Melodia. State of the art in protocol research for underwater acoustic sensor networks. 11 2006.
- [2] The World Bank. What is the blue economy?, June 6 2017.
- [3] Russell B. Wynn, Veerle A. I. Huvenne, Timothy P. Le Bas, Bramley J. Murton, Douglas P. Connelly, Brian J. Bett, Henry A. Ruhl, Kirsty J. Morris, Jeffrey Peakall, Daniel R. Parsons, Esther J. Sumner, Stephen E. Darby, Robert M. Dorrell, and James E. Hunt. Autonomous underwater vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, pages 451–468, 2014.
- [4] Stephanie Petillo and Henrik Schmidt. Exploiting adaptive and collaborative AUV autonomy for detection and characterization of internal waves. *IEEE Journal of Oceanic Engineering*, pages 150–164, 2014.
- [5] R. Onken, Heinz-Volker Fiekas, L. Béguey, I. Borrione, A. Funk, M. Hemming, J. Hernández-Lasheras, K. Heywood, J. Kaiser, M. Knoll, B. Mourre, P. Oddo, P. Poulain, B. Queste, A. Russo, K. Shitashima, M. Siderius, and Elizabeth T. Küsel. High-resolution observations in the western mediterranean sea: The REP14-MED experiment. *Ocean Science*, pages 321–335, 2016.
- [6] Pere Ridao et al. Intervention AUVs: The next challenge. *Annual Reviews in Control*, pages 227–241, 2015.
- [7] Scott Reed, Jon Wood, and Chris Haworth. The detection and disposal of IED devices within harbor regions using AUVs, smart ROVs and data processing/fusion technology. In *Proceeding of the 2010 International WaterSide Security Conference*, pages 1–7, 2010.
- [8] N. M. Abreu and A. Matos. Minehunting mission planning for autonomous underwater systems using evolutionary algorithms. *Unmanned Systems*, pages 323–349, 2014.
- [9] Filippo Campagnaro, Alberto Signori, and Michele Zorzi. Wireless Remote Control for Underwater Vehicles. *Journal of Marine Science and Engineering*, page 736, October 2020.
- [10] Milica Stojanovic and James Preisig. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE Communications Magazine*, 47(1):84–89, 2009.

- [11] X. Su, I. Ullah, X. Liu, and D. Choi. A review of underwater localization techniques, algorithms, and challenges. *Journal of Sensors, special issue on Sensors and Applications in Agricultural and Environmental Monitoring*, 2020:1–24, January 13 2020.
- [12] Jiangying Qin, Ming Li, Deren Li, Jiageng Zhong, and Ke Yang. A survey on visual navigation and positioning for autonomous uuv's. *Remote Sensing*, 14(15), 2022.
- [13] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [14] Philipp Woock. Survey on suitable 3d features for sonar-based underwater navigation. In *2012 Oceans - Yeosu*, pages 1–6, 2012.
- [15] Melike Erol-Kantarci, Hussein T. Mouftah, and Sema Oktug. A survey of architectures and localization techniques for underwater acoustic sensor networks, 2011.
- [16] Shuanglong Liu, Chun Zhang, and Yu Huang. Research on acoustic source localization using time difference of arrival measurements. In *Proceedings of 2012 International Conference on Measurement, Information and Control*, volume 1, pages 220–224, 2012.
- [17] Filip Mandić, Nikola Mišković, and Ivan Lončar. Underwater acoustic source seeking using time-difference-of-arrival measurements. *IEEE Journal of Oceanic Engineering*, 45(3):759–771, 2020.
- [18] Kenneth V Mackenzie. Nine-term equation for sound speed in the oceans. *The Journal of the Acoustical Society of America*, 70(3):807–812, 1981.
- [19] Simo Särkkä. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.
- [20] H.-A. Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, 2004.
- [21] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. isam: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008.
- [22] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *2011 IEEE International Conference on Robotics and Automation*, pages 3281–3288, 2011.
- [23] J. González-García, A. Gómez-Espinosa, E Cuan-Urquizo, L. G. García-Valdovinos, T. Salgado-Jiménez, and J. A. E. Cabello. Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions. *Applied Science, special issue on Underwater Robots in Ocean and Coastal Applications*, 10(4):1–37, February 13 2020.

- [24] R. Es-sadaoui, M. Guermoud, J. Khallaayoune, and T. Brizard. Autonomous underwater vehicles navigation and localization systems: A survey. In *Proceeding of SADASC 2018*, pages 1–6, May 28 2018.
- [25] P. H. Milne. *Underwater Acoustic Positioning Systems*. Spon Press, 1983.
- [26] J. Shi and Hy. Song. Research on techniques to improve accuracy in sbl locating systems. *J. Marine. Sci. Appl.*, 7:59–64, 2008.
- [27] Gianni Cario, Alessandro Casavola, Gianfranco Gagliardi, Marco Lupia, and Umberto Severino. Accurate localization in acoustic underwater localization systems. *Sensors*, 21(3), 2021.
- [28] Tongwei Zhang, Baohua Liu, and Yeyao Liu. Positioning systems for jiaolong deep-sea manned submersible: Sea trial and application. *IEEE Access*, 6:71644–71650, 2018.
- [29] Qinghua Luo, Xiaozhen Yan, Chunyu Ju, Yunsai Chen, and Zhenhua Luo. An ultra-short baseline underwater positioning system with kalman filtering. *Sensors*, 21(1), 2021.
- [30] Marco Morgado, Paulo Oliveira, and Carlos Silvestre. Robust outliers detection and classification for usbl underwater positioning systems. In *CONTROLO’2014 – Proceedings of the 11th Portuguese Conference on Automatic Control*, pages 555–565, 2015.
- [31] Arkadiusz Tomczak, Grzegorz Stępień, Tomasz Abramowski, and Artur Bejger. Subsea wellhead spud-in marking and as-built position estimation method based on ultra-short baseline acoustic positioning. *Measurement*, 195:111155, 2022.
- [32] Pedro Batista, Carlos Silvestre, and Paulo Oliveira. Ges integrated lbl/usbl navigation system for underwater vehicles. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 6609–6614, 2012.
- [33] Eric Wolbrecht, David Pick, John Canning, and Dean Edwards. Improving auv localization accuracy by combining ultra-short-baseline and long-baseline measurements systems in a post-processing extended kalman filter. In *OCEANS 2019 MTS/IEEE SEATTLE*, pages 1–7, 2019.
- [34] Irene Tallini, Luca Iezzi, Petrika Gjanci, Chiara Petrioli, and Stefano Basagni. Localizing Autonomous Underwater Vehicles: Experimental Evaluation of a Long Baseline Method. In *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 443–450, 2021.
- [35] Luca Iezzi, Chiara Petrioli, and Stefano Basagni. An Adaptive Extended Kalman Filter for State and Parameter Estimation in AUV Localization. In *ICC 2023 - IEEE International Conference on Communications*, pages 3932–3938, 2023.
- [36] Team: Tecnologie emergenti per l’archeologia marina. <https://www.teamarcheo.it/>.

- [37] Jonathan Gelli, Alessia Meschini, Niccolò Monni, Marco Pagliai, Alessandro Ridolfi, Lorenzo Marini, and Benedetto Allotta. Design and testing of a compact autonomous underwater vehicle for archaeological surveying and monitoring. In *Proceedings of IEEE/OES AUV 2018*, pages 1–6, 2018.
- [38] J. Gelli, A. Meschini, N. Monni, M. Pagliai, A. Ridolfi, L. Marini, and B. Allotta. Development and design of a compact autonomous underwater vehicle: Zeno AUV. *IFAC-PapersOnLine*, 51(29):20–25, 2018. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018.
- [39] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng. ROS: An open-source Robot Operating System. In *Proceeding of the ICRA Workshop on Open Source Software*, volume 3, pages 1–5, 2009.
- [40] C. Petrioli, R. Petroccia, J. R. Potter, and D. Spaccini. The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks. *Elsevier Ad Hoc Networks*, 34(C):224–238, November 2015.
- [41] Zhao Li, Stan E. Dosso, and Dajun Sun. Motion-compensated acoustic localization for underwater vehicles. *IEEE Journal of Oceanic Engineering*, 41(4):840–851, 2016.
- [42] Y. Han, C. Zheng, and D. Sun. Accurate underwater localization using LBL positioning system. In *Proceedings of MTS/IEEE OCEANS 2015*, pages 1–4, 2015.
- [43] Yiqun Wang, Chunhui Xu, Huixi Xu, Hongyu Zhao, and Jian Liu. An integrated navigation algorithm for AUV based on pseudo-range measurements and error estimation. In *Proceedings of IEEE ROBOT 2016*, pages 1625–1630, 2016.
- [44] Dajun Sun, Cuie Zheng, Miao Yu, and Yixian Shuai. Research and implementation on multi-beacon aided auv integrated navigation algorithm based on ukf. *ICVIS 2018*, 2018.
- [45] Wanlong Zhao, Shuyin Zhao, Gongliang Liu, and Weixiao Meng. Range-only single beacon based multisensor fusion positioning for auv. *IEEE Sensors Journal*, 23(19):23399–23409, 2023.
- [46] Tao Zhang, Hongfei Shi, Liping Chen, Yao Li, and Jinwu Tong. Auv positioning method based on tightly coupled sins/lbl for underwater acoustic multipath propagation. *Sensors*, 16(3), 2016.
- [47] Marco Bernardi, Christian Cardia, Petrika Gjanci, Andrea Monterubbiano, Chiara Petrioli, Luigi Picari, and Daniele Spaccini. The diver system: Multimedia communication and localization using underwater acoustic networks. In *Proceeding of IEEE WoWMoM 2019*, pages 1–8, 2019.
- [48] Laszlo Techy, Kristi A. Morganseny, and Craig A. Woolseyz. Long-baseline acoustic localization of the Seaglider underwater glider. In *Proceedings of the 2011 American Control Conference*, pages 3990–3995, 2011.

- [49] Eric Wolbrecht, David Pick, John Canning, and Dean Edwards. Improving AUV localization accuracy by combining ultra-short-baseline and long-baseline measurements systems in a post-processing extended Kalman filter. In *Proceedings of MTS/IEEE OCEANS 2019*, pages 1–7, 2019.
- [50] Sandipa Singh, Matthew Grund, Brian Bingham, Ryan Eustice, Hanumant Singh, and Lee Freitag. Underwater acoustic navigation with the WHOI Micro-Modem. In *Proceedings of MTS/IEEE OCEANS 2006*, pages 1–4, 2006.
- [51] I. Tallini, L. Iezzi, P. Gjanci, C. Petrioli, and S. Basagni. Localizing autonomous underwater vehicles: Experimental evaluation of a long baseline method. In *Proceedings of IEEE WCNEE 2021*, pages 443–450, Virtual Conference, July 16 2021.
- [52] Nilanjan Patra and Smita Sadhu. Adaptive extended Kalman filter for the state estimation of anti-lock braking system. In *Proceedings of IEEE INDICON 2015*, pages 1–6, New Delhi, India, December 17–20 2015.
- [53] Yulong Huang, Yonggang Zhang, Zhemin Wu, Ning Li, and Jonathon Chambers. A novel adaptive Kalman filter with inaccurate process and measurement noise covariance matrices. *IEEE Transactions on Automatic Control*, 63(2):594–601, February 2018.
- [54] Aritro Dey, Manasi Das, Smita Sadhu, and Tapan Kumar Ghoshal. Adaptive divided difference filter for parameter and state estimation of non-linear systems. *IET Signal Processing*, 9(4):369–376, 2015.
- [55] Shahrokh Akhlaghi, Ning Zhou, and Zhenyu Huang. Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation. In *Proceeding of IEEE PES General Meeting 2017*, pages 1–5, Chicago, IL, July 16–20 2017.
- [56] Junting Wang, Tianhe Xu, and Zhenjie Wang. Adaptive robust unscented Kalman filter for AUV acoustic navigation. *Sensors*, 20(1):1–16, December 2019.
- [57] Yulin Xu et al. USBL positioning system based adaptive Kalman filter in AUV. In *Proceedings of MTS/IEEE OCEANS 2018–Kobe Techno-Oceans OTO*, pages 1–4, Kobe, Japan, May 28–31 2018.
- [58] Xinhui Shao, Bo He, Jia Guo, and Tianhong Yan. The application of AUV navigation based on adaptive extended Kalman filter. In *Proceedings of MTS/IEEE OCEANS 2016*, pages 1–4, Shanghai, China, April 10–13 2016.
- [59] Chingiz Hajiyev and Halil Ersin Soken. Q-adaptation of UKF algorithm for estimation of the autonomous underwater vehicles dynamics. In *Proceedings of CDSR 2018*, Niagara Falls, Canada, June 7–10 2018.
- [60] Yu Shan Sun, Wen Jiang Li, Zai Bai Qin, Hong Li Chen, and Ji Qing Li. Application of modified self-adaptive Kalman filter in integrated navigation system of autonomous underwater vehicle. *Applied Mechanics and Materials*, 79:298–303, July 2011.

- [61] Yiqun Wang, Chunhui Xu, Huixi Xu, Hongyu Zhao, and Jian Liu. An integrated navigation algorithm for auv based on pseudo-range measurements and error estimation. In *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1625–1630, 2016.
- [62] Ge Zhang, Guoxing Yi, Zhennan Wei, Yangguang Xie, and Ziyang Qi. A novel positional calibration method for an underwater acoustic beacon array based on the equivalent virtual long baseline positioning model. *Journal of Marine Science and Engineering*, 12(5), 2024.
- [63] Wanlong Zhao, Huifeng Zhao, Gongliang Liu, and Guoyao Zhang. Anfis-ekf-based single-beacon localization algorithm for auv. *Remote Sensing*, 14(20), 2022.
- [64] Josep Aulinas, Yvan Petillot, Joaquim Salvi, and Xavier Llado. The SLAM problem: A survey. In *Proceedings of CCIA 2008*, pages 363–371, Sant Martí d’Empúries, Spain, October 22–24 2008.
- [65] J. Vaganay, P. Baccou, and B. Jouvencel. Homing by acoustic ranging to a single beacon. In *Proceedings of MTS/IEEE OCEANS 2000*, volume 2, pages 1457–1462, Providence, RI, September 11–14 2000.
- [66] Edwin Olson, John J. Leonard, and Seth Teller. Robust range-only beacon localization. *IEEE Journal of Oceanic Engineering*, 31(4):949–958, October 2006.
- [67] P. Vaz Teixeira, M. B. Nogueira, and J. B. Sousa. Long baseline beacon position estimation. In *Proceedings of MTS/IEEE OCEANS 2011–Spain*, pages 1–7, Santander, Spain, June 6–9 2011.
- [68] P. Newman and J. Leonard. Pure range-only sub-sea SLAM. In *Proceedings of IEEE ICRA 2003*, volume 2, pages 1921–1926, Taipei, Taiwan, September 14–19 2003.
- [69] E. Rohmer, S. P. N. Singh, and M. Freese. CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013.
- [70] Olivier Michel. Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*, page 5, 2004.
- [71] Carlo Pinciroli, Vito Trianni, Rehan O’Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Timothy Stirling, Álvaro Gutiérrez, Luca Maria Gambardella, and Marco Dorigo. ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotics. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5027–5034, 2011.
- [72] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning, 2021.

- [73] Andr as Varga and Rudolf Hornig. An Overview of the OMNeT++ Simulation Environment. In *Proceedings of the 1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems*, 2010.
- [74] George F Riley and Thomas R Henderson. The ns-3 network simulator. In *Modeling and tools for network simulation*. Springer, 2010.
- [75] Nikhil Handigol, Brandon Heller, Vimalkumar Jeyakumar, Bob Lantz, and Nick McKeown. Reproducible network experiments using container-based emulation. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, page 253–264, 2012.
- [76] Chiara Petrioli, Roberto Petrocchia, John R. Potter, and Daniele Spaccini. The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks. *Ad Hoc Networks*, 34:224–238, 2015.
- [77] ns-3 Model Library, UAN Framework. Last accessed: Feb. 2, 2025.
- [78] Riccardo Masiero, Saiful Azad, Federico Favaro, Matteo Petrani, Giovanni Toso, Federico Guerra, Paolo Casari, and Michele Zorzi. DESERT Underwater: An NS-Miracle-based framework to design, simulate, emulate and realize test-beds for underwater network protocols. In *Oceans*, pages 1–10, 2012.
- [79] Robert Martin, Sanguthevar Rajasekaran, and Zheng Peng. Aqua-Sim Next Generation: An NS-3 Based Underwater Sensor Network Simulator. In *Proceedings of the International Conference on Underwater Networks & Systems, WUWNET’17*, 2017.
- [80] Abdelhakim Amer, Olaya  lvarez-Tu  n, Halil Ibrahim Ugurlu, Jonas le Fevre Sejersen, Yury Brodskiy, and Erdal Kayacan. UNav-Sim: A Visually Realistic Underwater Robotics Simulator and Synthetic Data-generation Framework, 2023.
- [81] Mario Prats, Javier P  rez, J. Javier Fern  ndez, and Pedro J. Sanz. An Open Source Tool for Simulation and Supervision of Underwater Intervention Missions. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2577–2582, 2012.
- [82] Musa Morena Marcusso Manh  es, Sebastian A. Scherer, Martin Voss, Luiz Ricardo Douat, and Thomas Rauschenbach. UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–8, 2016.
- [83] Mabel M. Zhang, Woen-Sug Choi, Jessica Herman, Duane Davis, Carson Vogt, Michael McCarrin, Yadunund Vijay, Dharini Dutia, William Lew, Steven Peters, and Brian Bingham. DAVE Aquatic Virtual Environment: Toward a General Underwater Robotics Simulator. In *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, pages 1–8, 2022.

- [84] Ivan Lončar, Juraj Obradović, Natko Kraševac, Luka Mandić, Igor Kvasić, Fausto Ferreira, Vladimir Slošić, Đula Nađ, and Nikola Mišković. MARUS - A Marine Robotics Simulator. In *OCEANS 2022, Hampton Roads*, pages 1–7, 2022.
- [85] Atsushi Suzuki, Kazuyuki Masutomi, Isao Ono, Hideaki Ishii, and Takashi Onoda. CPS-Sim: Co-Simulation for Cyber-Physical Systems with Accurate Time Synchronization. *IFAC-PapersOnLine*, 2018.
- [86] Miguel Calvo-Fullana, Daniel Mox, Alexander Pyattaev, Jonathan Fink, Vijay Kumar, and Alejandro Ribeiro. ROS-NetSim: A Framework for the Integration of Robotic and Network Simulators. *IEEE Robotics and Automation Letters*, pages 1120–1127, 2021.
- [87] Srikrishna Acharya, Amrutur Bharadwaj, Yogesh Simmhan, Aditya Gopalan, Parimal Parag, and Himanshu Tyagi. CORNET: A Co-Simulation Middleware for Robot Networks. In *2020 International Conference on COMMunication Systems & NETWORKS (COMSNETS)*, pages 245–251, 2020.
- [88] Sabur Baidya, Zoheb Shaikh, and Marco Levorato. FlyNetSim: An Open Source Synchronized UAV Network Simulator based on ns-3 and Ardupilot. In *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 37–45, 2018.
- [89] Fabio D’Urso, Corrado Santoro, and Federico Fausto Santoro. An Integrated Framework for the Realistic Simulation of Multi-UAV Applications. *Computers & Electrical Engineering*, 74:196–209, 2019.
- [90] Emon Dey, Jumman Hossain, Nirmalya Roy, and Carl Busart. SynchroSim: An Integrated Co-simulation Middleware for Heterogeneous Multi-robot System. In *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 334–341, 2022.
- [91] Emon Dey, Mikolaj Walczak, Mohammad Saeid Anwar, Nirmalya Roy, Jade Freeman, Timothy Gregory, Niranjani Suri, and Carl Busart. A Novel ROS2 QoS Policy-Enabled Synchronizing Middleware for Co-Simulation of Heterogeneous Multi-Robot Systems. In *2023 32nd International Conference on Computer Communications and Networks (ICCCN)*, pages 1–10, 2023.
- [92] Fabian Steinmetz, Daniel A Duecker, Nils Sichert, Christian Busse, Edwin Kreuzer, and Bernd-Christian Renner. UWRange: An Open ROS Framework for Simulating Acoustic Ranging and Localization for Underwater Robots under Realistic Conditions. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8442–8449, 2022.
- [93] The Network Simulator - ns-2. Last accessed: Feb. 2, 2025.
- [94] Nicola Baldo, Federico Maguolo, Marco Miozzo, Michele Rossi, and Michele Zorzi. ns2-MIRACLE: a modular framework for multi-technology and cross-layer support in network simulator 2. In *1st International ICST Workshop on Network Simulation Tools*, 2010.

-
- [95] Dirk Merkel et al. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*, 239(2):2, 2014.
 - [96] Federico Guerra, Paolo Casari, and Michele Zorzi. World ocean simulation system (WOSS) a simulation tool for underwater networks with realistic propagation modeling. In *Proceedings of the Fourth ACM International Workshop on UnderWater Networks*, pages 1–8, 2009.
 - [97] Emanuele Giona. ns3-ext-process.