UNIVERSITÀ
DEGLI STUDI
FIRENZE

Development and testing of multisensor strategies for autonomous underwater navigation and mapping

*A dissertation submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy
in*

INDUSTRIAL ENGINEERING
Scientific Disciplinary Sector ING-IND/13

Candidate
Alessandro Bucci

Advisor
Prof. Benedetto Allotta

Co-advisor
Dr. Alessandro Ridolfi

Coordinator
Prof. Giovanni Ferrara
Giovanni Ferrara

Department of Industrial Engineering
University of Florence
Florence, Italy

XXXV Ph.D. Cycle - Years 2019/2022

To my family

Tons of people have my gratitude for their contribution to my experience at the University of Florence during the last three years.
Firstly, I would like to express my gratitude to Prof. Benedetto Allotta that introduced me to the world of robotics. Thanks to him, I had the opportunity to work in this field for three years, making a long-lasting wish come true.
Special thanks go to my co-advisor, Dr. Alessandro Ridolfi, for his advice, time, and guidance. I would like to sincerely thank you for trusting me during these years and for your precious help in achieving the results reported in this thesis.
"Teamwork makes the dream work!"; a huge thanks goes to my lab mates Lorenzo, Leonardo, Alberto, Edoardo, Matteo, Jonathan, Andrea, Nicola, Chiara, Guido, Gherardo and Mirco. The results of this thesis wouldn't have been possible without them. More friends than just colleagues, they have made these three years stimulating. I want to thank them for their support, for all they have taught me, and for all the fun we have had in these years.
My heartfelt thanks go to my parents and family, who supported and encouraged me during this challenging period. Mom and Dad are the ones who made it possible to start and conclude this journey.
Finally, I would like to thank all my friends, who have made these years great and amusing.
Thanks to all the people mentioned here for shaping the person I am!

Alessandro Bucci. University of Florence, January 2023.

IN the underwater domain, guaranteeing accurate navigation for an Autonomous Underwater Vehicle (AUV) is a complex but fundamental task to be achieved. As a matter of fact, only by ensuring a correct AUV localization it is possible to accomplish surveillance, monitoring, and inspection missions.

Firstly, focusing on the attitude estimation filters, a strategy based on Extended Kalman Filter (EKF) and Lie groups theory for AUVs orientation initialization has been developed to face the presence of magnetic disturbances, which makes magnetometer measurements unreliable. The procedure is performed when the vehicle is on the sea surface to replace the magnetometer measurements and to evaluate the offset caused by the unknown disturbances. The strategy has been validated and evaluated with real data acquired during experimental campaigns at sea.

Turning to position estimation algorithms, most of the navigation filters for AUVs are based on Bayesian estimators, such as the Kalman Filter (KF), the EKF, or the Unscented Kalman Filter (UKF), and employ different instruments, often including the Doppler Velocity Log (DVL) to perform the localization task. Recently, the use of payload sensors, such as cameras or Forward Looking SONARs (FLSs), in navigation-aiding has arisen as an interesting research field in the attempt to reduce localization error drift. Such sensors, if used simultaneously, can provide multiple observations, which can be combined in a Kalman filtering framework to increase navigation robustness against noise sources. Navigation techniques that employ multiple devices can provide a high improvement in the estimation quality, but they can also cause an increase in terms of computational load. Consequently, strategies representing a trade-off between these two conflicting goals have been investigated. Using an augmented set of devices able to provide navigation information represents an intrinsic boost in redundancy: DVL-denied scenarios, such as very close to the seafloor or other surfaces or when a substantial number of gaseous bubbles is present, could thus be managed.

Two different UKF-based frameworks have been implemented and compared: on the one hand, a centralized iterative UKF-based navigation approach and on the other hand, a sensor fusion framework with parallel local UKFs. To the author's best knowledge, the fusion of inertial, acoustic, and optical data in a UKF algorithm and the application of the presented sensor fusion strategies to AUV navigation is novel. It is necessary to highlight that while the centralized strategy guarantees the best improvements in terms of estimation quality, decentralized strategies provide an increase of robustness against measurement reduction.

To overcome the limitation of the Kalman filtering strategies and to accomplish

the mapping task during vehicle navigation, a factor graph-based Simultaneous Localization And Mapping (SLAM) strategy, which relies on DVL and optical measurements, has been developed and tested. The strategy has been implemented to fuse the data coming from these onboard sensors and to provide a reconstruction of the seabed inspected by the AUV.

The proposed solutions have been firstly validated with realistic simulations made through the Unmanned Underwater Vehicle Simulator (UUV Simulator), where a dynamic model of FeelHippo AUV was implemented. Moreover, both the UKF and the SLAM strategies have been tested with real experimental data acquired during several experimental campaigns at sea.

Operating on both the orientation and the position estimation filters, the presented works propose advances to improve underwater vehicles' localization and mapping capabilities.

Contents

List of Figures

List of Tables

Acronyms

AHRS Attitude and Heading Reference System

AIO Acoustic Inertial Odometry

AKAZE Accelerated-KAZE

AO Acoustic Odometry

ARCHEOSUb Autonomous underwater Robotic and sensing systems for Cultural HEritage discovery COnservation and in sitU valorization

ARROWS ARcheological RObot systems for the World's Seas

ATR Automatic Target Recognition

AUV Autonomous Underwater Vehicle

BOOMER reconfiguraBle autOnOmous systeMs for undErwater waRfare

CB Center of Buoyancy

CG Center of Gravity

CPU Central Processing Unit

CSSN Naval Support and Experimentation Center (Centro di Supporto e Sperimentazione Navale)

DIEF Department of Industrial Engineering in Florence

DLT Direct Linear Transformation

DOF Degree of Freedom

DR Dead Reckoning

DS Depth Sensor

DVL Doppler Velocity Log

EKF Extended Kalman Filter

ERL European Robotics League

EUMR EUMarineRobots

FLS Forward Looking SONAR

FOG Fiber Optic Gyroscope

GNSS Global Navigation Satellite System

GPS Global Positioning System

GTSAM Georgia Tech Smoothing And Mapping

IKF Information Kalman Filter

IMU Inertial Measurement Unit

iSAM2 Incremental Smoothing and Mapping 2

ISME Interuniversity Center of Integrated Systems for the Marine Environment

KF Kalman Filter

LG-EKF Extended Kalman Filter on Lie groups

MAP Maximum A Posteriori

MDM Lab Mechatronics and Dynamic Modeling Laboratory

NED North, East, Down

ORB Oriented FAST and Rotated BRIEF

PASSport Operational Platform managing a fleet of semi-autonomous drones exploiting GNSS high Accuracy and Authentication to improve Security Safety in port areas

PCL Point Cloud Library

RAMI Robotics for Asset Maintenance and Inspection

RANSAC RAndom SAmple Consensus

ROS Robot Operating System

ROV Remotely Operated Vehicle

RPY Roll, Pitch and Yaw

RV Random Variable

SAUC-E Student Autonomous Underwater Vehicles Challenge -Europe

SIFT Scale Invariant Feature Transform

SLAM Simultaneous Localization And Mapping

SNAME Society of Naval Architects and Marine Engineers

SURF Speeded Up Robust Features

THESAURUS TecnicHe per l'Esplorazione Sottomarina Archeologica mediante l'Utilizzo di Robot aUtonomi in Sciami

UAV Unmanned Aerial Vehicle

UKF Unscented Kalman Filter

UNIFI University of Florence

USBL Ultra-Short BaseLine

UT Unscented Transform

UUV Unmanned Underwater Vehicle

UUV Simulator Unmanned Underwater Vehicle Simulator

VIO Visual Inertial Odometry

VO Visual Odometry

Chapter 1

Introduction and thesis motivation

FROM geology to exploration and surveillance of archaeological sites and from Oil and Gas industry to reconnaissance for military purposes, exploring and understanding seas and oceans is a matter of primary importance. Considering their human hostile nature, since the 1960s, seas and oceans have been explored with the aid of robots. The first Unmanned Underwater Vehicles (UUVs) were teleoperated ones and are referred in the technical literature as Remotely Operated Vehicles (ROVs). A cable, usually called umbilical cable, acts as a constant connection providing power and communications, and specialised operators are thus able to control the vehicle using the feedback forwarded by the on-board sensors. In the last decades Autonomous Underwater Vehicles (AUVs), which are completely autonomous, have gained interest with respect to ROVs. Indeed, such vehicles do not require human intervention (except for deployment and recovery), are usually equipped with electric batteries, and possess dedicated systems used to control their motion. Since the demanded tasks of underwater vehicles have become more and more challenging [1], [2], researchers and scientists are following the tide of change and are pushing the boundaries of AUVs capabilities by integrating cutting-edge technologies. Indeed, autonomous inspection [3], and intervention [4] strategies for underwater installations, exploration planning solutions [5], and autonomous coverage approaches [6], have become essential tools to execute demanding and hazardous subsea operations. One of the most significant and complex tasks in autonomous underwater exploration is to retrieve the vehicle's pose within the surrounding environment, making use of precise and reliable navigation and localization systems, which are necessary regardless of the kind of mission or task the underwater vehicle is required to perform. In addition to this, perceptual devices (such as optical cameras and acoustic devices) able to sense the surrounding environment have been earning attention throughout the last decades to acquire data for monitoring and inspection purposes. The use of optical and acoustic equipment to aid navigation has emerged as a relevant alternative or support to traditional navigation sensors.

Consequently, the research carried out during the Ph.D. period aimed at pushing forward the use of multisensor approaches in underwater navigation-aiding, developing and validating a novel solution tailored to the underwater environment.

1.1 Overall framework

The research activity described in this work has been carried out at the Mechatronics and Dynamic Modeling Laboratory (MDM Lab), a research laboratory that belongs to the Department of Industrial Engineering in Florence (DIEF) of the University of Florence (UNIFI). UNIFI DIEF has been operative in the field of underwater robotics since 2010 thanks to the participation in the Tuscany-funded project TecnicHe per l'Esplorazione Sottomarina Archeologica mediante l'Utilizzo di Robot aUtonomi in Sciami (THESAURUS), where the main goal was the development of a swarm of AUVs for underwater archaeology. Since then, the MDM Lab research group has taken part in several cutting-edge national and international projects in the marine robotics field. In 2013, the MDM Lab joined as the coordinating partner of the European FP7 project ARcheological RObot systems for the World's Seas (ARROWS).

Since 2014, the University of Florence has joined the Interuniversity Center of Integrated Systems for the Marine Environment (ISME), which gathers Universities from all over Italy and whose main goal is to act as a common platform for joint operations for what concerns marine robotics and, generally speaking, marine field. As a result of the support of the Naval Support and Experimentation Center (Centro di Supporto e Sperimentazione Navale) (CSSN) of the Italian Navy (formalized as the SEALab joint laboratory), several tests at sea, with the aiming of trying out the vehicles of the UNIFI DIEF MDM Lab fleet, have been performed during the Ph.D. in La Spezia, Italy.

Together with the official spin-off company MDM Team S.r.l. of UNIFI

Figure 1.1: Logos of the main projects joined by the MDM Lab.

DIEF, the MDM Lab took part in the Autonomous underwater Robotic and sensing systems for Cultural HEritage discovery COnservation and in sitU valorization (ARCHEOSUb) European project in 2018 to realize technologies for underwater archaeology. The group joined the H2020 EUMarineRobots (EUMR) and Operational Platform managing a fleet of semi-autonomous drones exploiting GNSS high Accuracy and Authentication to improve Security Safety in port areas (PASSport) projects in 2019. While the EUMR project opens transnational access to national and regional marine robotics research infrastructures to all European researchers, ensuring their optimal use and joint development to establish a world-class marine robotics integrated infrastructure, the PASSport project aims to improve security in port areas through the combined use of Unmanned Aerial Vehicles (UAVs) and AUVs. In the framework of the EUMR project, underwater missions were performed by UNIFI DIEF at Vulcano Island, Messina (Italy) with FeelHippo AUV. The main goal was to detect carbon dioxide bubbles in the sea, which are responsible for negative effects on underwater fauna and flora. Several data employed to reach the main goals of this thesis have been acquired during this experimental campaign.
In 2020, the MDM Lab joined the military research project reconfiguraBle autOnOmous systeMs for undErwater waRfare (BOOMER), whose main goal is to perform acoustic measurements in challenging underwater environment through the employment of a fleet composed of autonomous underwater vehicles and autonomous surface vessels. In the context of this project, the MDM Lab realizes

a gateway buoy for underwater localization and a module to perform precise navigation tasks by employing high-quality cameras.
Finally, in 2021, the PATHfinder project proposes a solution based on a fleet of multi-domain (aerial, terrestrial and underwater) drones for emergency scenario management, where the MDM Lab manages the underwater inspection with FeelHippo AUV. To emphasize the potential utility of the PATHfinder proposed solution, the case study of Stromboli Island, Messina (Italy) is proposed as an optimal case history of multi-hazard occurrences that impacted the communities and the territory. This experimental campaign, carried out in September 2022, has been employed to perform the final tests of the main navigation strategies developed during the Ph.D. period.

ISME
Integrated Systems for Marine Environment
SEA Lab
SISTEMI ETEROGENEI AUTONOMI MARINI

Figure 1.2: Logos of ISME and SEALab.

In addition to the participation in regional, national, and European projects, UNIFI DIEF took part in many non-student and student robotics competitions during the last few years. Just to name a few examples, a team of UNIFI DIEF participated to Student Autonomous Underwater Vehicles Challenge - Europe (SAUC-E) competition in 2012, 2013, 2016, and European Robotics League (ERL) SAUC-E in 2018, and 2019. Furthermore, in 2015 euRathlon competition was joined, whereas a team took part in ERL in 2017. Throughout the years, the teams of UNIFI DIEF have gained expertise and proficiency. In 2013 and 2016, the third place was obtained, while in 2017, the participating team was awarded Second-in-Class in “Pipe inspection and search for search for missing workers (Sea+Air)” during ERL Emergency Robots 2017. In 2018 and 2019, the team won the competition in the sea domain, and it was awarded “Best Marine Team” during ERL SAUC-E 2018, and ERL SAUC-E 2019. Finally, in 2022, the team reached the maximum number of points during the tests at sea at the international competition Robotics for Asset Maintenance and Inspection (RAMI).

1.2 FeelHippo AUV

FeelHippo AUV, whose last version is depicted in Fig. 1.3, is a compact and lightweight AUV developed by the MDM Lab in 2013 for joining student and non-student robotics competitions [7]. In recent years, it has faced several hardware and software upgrades that have made FeelHippo AUV a reliable and versatile platform for conducting research activities and project-related tasks. FeelHippo AUV is composed of a central body made of a Plexiglass® hull, where the hardware and electronics are housed, and two aluminum pipes attached under the main body, which contain the batteries. At the time of writing, FeelHippo AUV propulsion is realized through six thrusters (two at the stern, two at the bow, and one each on both sides inclined at 45°) arranged in a vectored configuration that allows for the control of all the Degree of Freedom (DOF) of the vehicle, except for the pitch motion that depends entirely on the inner mass distribution and the vehicle configuration. FeelHippo AUV main characteristics are summarized in Tab. 1.1.

Table 1.1: FeelHippo AUV physical data and performance.

FeelHippo AUV main features	
Dry mass [kg]	35
Dimensions [m]	$0.6 \times 0.64 \times 0.5$
Controlled DOFs	5
Thrusters	6
Battery life [h]	4
Maximum depth [m]	30.0
Maximum longitudinal speed [m/s]	1.0

The list of the primary electronic components and the sensor sets with which FeelHippo AUV is equipped are reported below:

- Intel i7-based LP-175-Commel motherboard (main computer);
- NVIDIA Jetson Nano (payload computer);
- U-blox 7P precision Global Navigation Satellite System (GNSS), capable to acquire data from the satellites of the United States' Global Positioning System (GPS) and the European Galileo system;

- Orientus Advanced Navigation Attitude and Heading Reference System (AHRS);
- KVH DSP 1760 single-axis high precision Fiber Optic Gyroscope (FOG);
- Nortek DVL1000 Doppler Velocity Log (DVL), measuring linear velocity and acting as Depth Sensor (DS);
- EvoLogics S2CR 18/34 acoustic modem;
- Teledyne BlueView M900 2D Forward Looking SONAR (FLS);
- one Microsoft Lifecam Cinema forward-looking camera;
- one Microsoft Lifecam Cinema bottom-looking cameras;

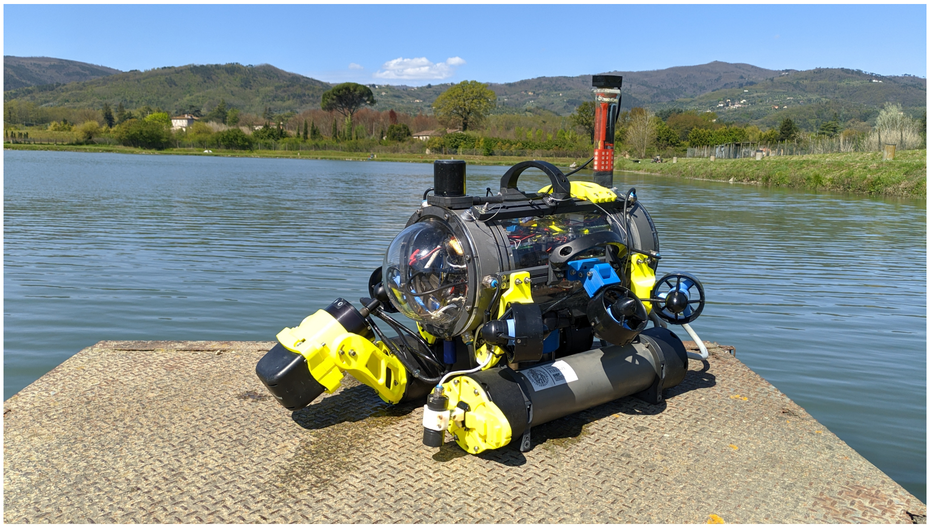


Figure 1.3: FeelHippo AUV before an on-field underwater mission.

The main computer (Intel i7-based LP-175-Commel motherboard) runs the software architecture based on the Robot Operating System (ROS) framework [8], and it is used for onboard processing of guidance, navigation, and control algorithms, for supervising the state of the vehicle, and for managing the communication channels. The payload computers, an NVIDIA Jetson Nano connected to the main computer, is exploited for running onboard machine learning algorithms and Automatic Target Recognition (ATR) solutions [9].

1.3 Thesis structure

This thesis is organized as follows.
Chapter 2 reviews the state-of-the-art in navigation strategies for the underwater domain. Kalman filtering approaches and Simultaneous Localization And Mapping (SLAM) algorithms based on inertial and payload sensors are analyzed. Chapter 2 also focuses on the main contributions of this thesis.
Chapter 3 is dedicated to the definition of the notation employed in this work, introducing the main theoretical concepts regarding AUV and sensor modeling and navigation strategies.
Chapter 4 introduces and discusses an orientation initialization algorithm to be employed in the presence of unknown magnetic disturbance. The proposed solution has been accurately validated during several experimental campaigns.
Chapter 5 is dedicated to the testing and comparison of Unscented Kalman Filter (UKF) solutions for multisensor navigation. A systematic analysis of the developed algorithms has been performed on both simulated and experimental data. All the proposed solutions have been tested in an all against all comparison.
In Chapter 6 advances for enhancing the performance of the position estimation filter are proposed. In particular, Chapter 6 introduces a SLAM strategy to employ the onboard sensors for navigation and mapping the surrounding environment. This strategy has been tested by exploiting simulated and experimental data to verify its performance in both navigation and mapping tasks. In particular, in the experimental results the main findings obtained during the Ph.D. period for the orientation and the position algorithms are fused in a new navigation framework.
Finally, Chapter 7 concludes the thesis by providing an analysis of the obtained results and discussing possible future developments.
Appendix A contains the description of developed automatic tools for point cloud post-processing. The data obtained from the SLAM algorithm can be elaborated to generate a simplified 3D geo-referenced textured reconstruction of the seabed.

Chapter 2

Related works and thesis contribution

THIS chapter reviews the related works that inspired this Ph.D. research work. Firstly, state-of-the-art in navigation strategies for the underwater domain are presented. Then, Maximum A Posteriori (MAP) and SLAM strategies that influenced the solution developed during the Ph.D. period are reviewed. Lastly, the main contribution of this thesis are highlighted.

2.1 State-of-the-art of Kalman filtering for underwater navigation

AUVs are now used for a wide variety of missions, and consequently, an accurate localization and navigation strategy is essential to ensure that objectives are achieved. AUV localization is a challenging task due to all the difficulties introduced by the underwater environment, such as the lack of GPS and the reduced visibility. As the simplest solution, Dead Reckoning (DR) strategies have proven satisfyingly reliable if the available sensors are sufficiently accurate. A common approach is to integrate over time measurements from a DVL combined with an attitude reference (obtained, e.g., from an Inertial Measurement Unit (IMU) together with a FOG), often leading to good navigational accuracy [10]. In some working scenarios, such as when the AUV is near to the seafloor or to other surfaces or when gaseous bubbles are present [11], the DVL operability results limited. Consequently, the development of sensor fusion strategies, capable of exploiting all the measurement information acquired by the AUV during its mission, has been an active and interesting research field. Focusing on underwater navigation applications, Visual Inertial Odometry (VIO) strategies have been appropriately investigated as a combination of inertial and optical data to overcome both sensors' limitations. This choice can be highly fruitful thanks to the complementarity of the employed sensor properties. While cameras suffer highly rapid movements and textureless environments, inertial sensors, as the IMUs, are independent of the observed scene and can be a useful support to the image-based navigation data. VIO strategies for underwater navigation have been addressed in [12] and [13]. While in [12] the feasibility of a stereo visual-inertial approach for AUV trajectory estimation is shown, the algorithm proposed in [13] combines information from an IMU in the form of linear accelerations and angular velocities, depth data from a DS, and feature tracking from a monocular downward facing camera to estimate the pose of the vehicle. Following the approaches introduced in the VIO field, Acoustic Inertial Odometry (AIO) has been explored in the last few years but still is in a more embryonic stage; some promising results can be found in [14], and [15]. On the one hand, the approach proposed in [14] efficiently fuses the acoustic observations from a 2D imaging SONAR and the inertial measurements from an IMU; on the other hand, the strategy presented in [15] is dedicated to the development of an underwater navigation system that does not rely on a DVL and where linear speed estimations are obtained exploiting data from a FLS. A more elaborate approach for both position and attitude estimation, usually exploited in marine robotics, is based on the Kalman Filter (KF) [16], on the Extended Kalman Filter (EKF) [17], [18] or the UKF [19], [20], [21] typically employed when nonlinearities in the dynamic description of the system arise. Generally speaking, all require knowledge of the linear speed of the AUV, which is (for the majority of times) obtained using specialized and underwater sensors, such as the DVL, or exploiting optical or acoustic payload. It is worth nothing that in the AUV navigation context, fusion strategy for redundant information as in the presence of acoustic, optical and inertial sensors is still a niche and open problem; preliminary works can be found in [22]. It is necessary to notice that in the algorithm proposed in [22], the AUV switches the utilizing sensor in appropriate case in real-time, and always navigates with the most useful sensor. A feature-based reliability value of each acquired optical and acoustic image is defined to decide which strategy can be employed, but the estimated speed values are not fused in a unique framework and only a switching strategy is adopted.

Turning to fusion strategies in the presence of redundant information, a first categorization can be made between centralized and decentralized methods. Historically, the former solutions represent the standard approach, and they advocate the usage of a single central filter devoted to estimation (a detailed discussion can be found in [23] and [24]). When compared with centralized solutions, decentralized methods significantly reduce the computational effort and increase the fault-tolerant capability of the overall navigation system [25], [26], [27], and [28]. Within the decentralized filtering realm, the federated filters introduced by Carlson in [25] and [26] represent one of the first developed solutions, which have been initially tested through simulations. In [27] and [28], some experimental results show that applying a fusion algorithm based on Kalman filtering can avoid the limitations of a single sensor, reduce its uncertainty impact and increase the confidence level of data. Besides, Information Kalman Filters (IKFs) are fused in a central processor unit in [29], [30]. Finally, consensus-based works [31], [32] typically involve a sensor network constituted of several AUV platforms [33], [34], but can also be considered in a single vehicle context.

As a consequence, historically, the vast majority of contributions concerning fusion strategies in navigation algorithms have been focused on constraining the navigational drift of the AUV; thus, particular emphasis has been paid to the use of multiple devices complementary to each other.

2.2 Related works on MAP and SLAM strategies for underwater navigation and mapping

Many estimation problems in robotics have an underlying optimization problem [35]. In most of these optimization problems, the objective to be maximized or minimized is composed of many different factors (e.g., a GPS measurement is applied to the pose of the vehicle at a particular time and can be referred as an unary factor, an IMU measurement can be related to two vehicle states at adjacent times and can represent an odometry factor). The use of factorial graphs in the design of algorithms for robotic applications has three main advantages. First, since many optimization problems in robotics have the property of locality, factorial graphs can model a wide variety of problems in all robotics domains, such as tracking, navigation, and mapping. Secondly, by clearly exposing the structure of the problem, reflection on factorial graphs offers many opportunities to improve the performance of key algorithms. Many classical algorithms can be viewed as the application of the elimination algorithm to a particular type of factorial graph. Still, this algorithm is only optimal for a small class of problems. In many applications, knowledge of the specific structure of the problem domain can improve the execution time of inference by orders of magnitude. Similarly, well-known algorithmic ideas from linear algebra can be generalized to factorial graphs, leading, for example, to incremental inference algorithms. Thirdly, apart from performance considerations, factorial graphs are useful when designing and thinking about how to model a problem, providing a common language to express ideas to collaborators and users of a particular algorithm. After working with factor graphs for a while, one begins to identify factor types as a particularly useful design unit. A factor type specifies how many variables a factor is related to and the semantics associated with the function to be calculated.

MAP estimation has recently become the standard approach for modern SLAM strategies [36]. Indeed, while fixed-lag smoothers and filtering solutions restrict the inference within a window of the latest states or to the latest state, respectively, MAP strategies estimate the entire history of the system by solving a nonlinear optimization problem. Both fixed-lag smoothers and filters marginalize older states, collapsing the corresponding information (usually) in a Gaussian prior. This approach can lead to reduced robustness against outlier data [37]. Since MAP strategies can quickly lead to an unsuitable approach for real-time applications, the development of incremental smoothing techniques has arisen as the state-of-the-art approach. Such techniques can reuse previously calculated quantities when new measurements or variables are added [38], [39]. In particular, in [39] a Bayes tree data structure is employed to perform incremental optimization on the factor graph. Also, the adopted solution possesses the ability to identify and update only a small subset of variables by accurately selecting the ones affected by the new measurement. A complete review can be found in [40] and the references therein.

Considering the underwater domain, two works have been taken as inspiration for the development of the factor graph employed in the proposed SLAM strategy. [41] proposes an algorithm to generate pose-to-pose constraints for pairs of SONAR images and to fuse these resulting pose constraints with the vehicle odometry in a pose graph optimization framework. In [42] Ultra-Short BaseLine (USBL) measurements are exploited as observations within the on-board navigation filter, where the localization task is solved as a MAP estimation problem. Both these solutions rely on Incremental Smoothing and Mapping 2 (iSAM2), which is the last evolution of the incremental smoothing and mapping solution developed in Georgia Tech Smoothing And Mapping (GTSAM). Furthermore, other graph-based SLAM strategies have been proposed to fuse the data obtained by the navigation sensors and the perception sensors, both acoustic and optical. In [43] this approach is used in an AUV for mine counter measurement and localization. While the graph is initialized by pose node from a GPS, a non-linear least square optimization is performed with the DVL and IMU-based DR estimations and the SONAR images. In [44] an acoustic structure from motion algorithm for recovering 3D scene structure from multiple 2D SONAR images while at the same time localizing the SONAR is presented.

Turning to visual SLAM, ORB-SLAM [45] is one of the most complete and simple algorithms, and the whole system is calculated around Oriented FAST and Rotated BRIEF (ORB) feature points, with features such as rotational scale invariance and fast detection. ORB-SLAM2 [46] is upgraded from ORB-SLAM, supporting monocular, binocular, and RGB-D modes, and has good adaptability. Finally, the latest ORB-SLAM3 [47] algorithm fuses optical images with inertial sensors. The excellent characteristics of the ORB-SLAM2 algorithm, which can achieve centimeter-level precision on the ground, represent an incentive for its application in underwater environments. Consequently, the visual part of the developed SLAM algorithm takes inspiration from the ORB-SLAM2 framework. Referring to the vision-based SLAM algorithm for underwater navigation and mapping, [48] addresses a visual mapping method for precise camera trajectory estimation and 3D reconstruction of underwater ship hull surface using a monocular camera as the primary sensor. [49] proposes an underwater visual SLAM system using a stereo camera, which has been tested in a circular pool.

Finally, an acoustic-visual-inertial SLAM strategy has been proposed in [50] and [51]. Data coming from a mechanical scanning SONAR, a stereo camera, and proprioceptive inertial sensors are fused in a tightly coupled non-linear optimization to estimate the vehicle trajectory and reconstruct the surrounding environment. There are few works where the DVL measurements are fused with other perception sensors in a SLAM strategies. In [52] a SLAM method, which uses a very sparse point cloud derived from a DVL to add constraints to a piecewise-planar framework, is proposed. A camera is also employed to bound drifts of odometry fused by a DVL, IMU and pressure DS [53]. Fiducial markers are also integrated into a visual SLAM framework with DVL, IMU, and DS in [54].

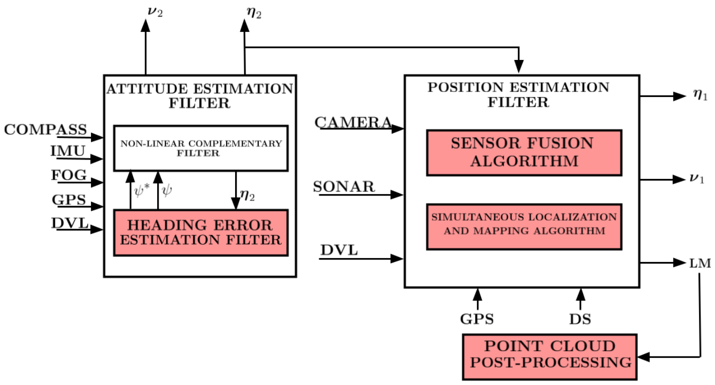

ν_2
η_2
ATTITUDE ESTIMATION FILTER
COMPASS
IMU
FOG
GPS
DVL
NON-LINEAR COMPLEMENTARY FILTER
ψ^*
ψ
η_2
HEADING ERROR ESTIMATION FILTER
POSITION ESTIMATION FILTER
CAMERA
SONAR
DVL
SENSOR FUSION ALGORITHM
SIMULTANEOUS LOCALIZATION AND MAPPING ALGORITHM
η_1
ν_1
LM
GPS
DS
POINT CLOUD POST-PROCESSING

Figure 2.1: Overall navigation filter framework, where the advances developed during the Ph.D. period for both the attitude and position estimation filters are depicted in red. The acronym LM stands for landmark and it indicates the landmark positions estimated with the SLAM algorithm.

2.3 Thesis contribution

The present work's main contribution focuses on developing navigation and mapping strategies that, exploiting all the available inertial, acoustic, and optical sensors, can guarantee a precise and robust localization of the AUV during the whole underwater mission (see Fig. 2.1 and Fig. 2.2). Working with a DVL, a monocular camera, and an FLS, more linear speed estimation could be obtained simultaneously to exclude possible outliers and to fuse the retrieved information optimally. Indeed, using multiple sensors to provide navigation information represents an intrinsic advantage in terms of redundancy, preventing failures due, for example, to underwater sensor-denied scenarios. DVL-denied scenarios are verified when the AUV is near to the seafloor or to other surfaces or when gaseous bubbles are present. While optical camera functioning is related to visibility conditions and texturing of the seafloor, acoustic sensors like FLSs are not influenced by these issues, but acoustic noise sources can reduce their operability.

Firstly, a strategy based on Extended Kalman Filter on Lie groups (LG-EKF) for AUVs orientation initialization in the presence of magnetic disturbances, making magnetometer measurements unreliable, has been developed. An initialization procedure to be performed when the vehicle is on the sea surface is necessary to replace the magnetometer measurements and to evaluate

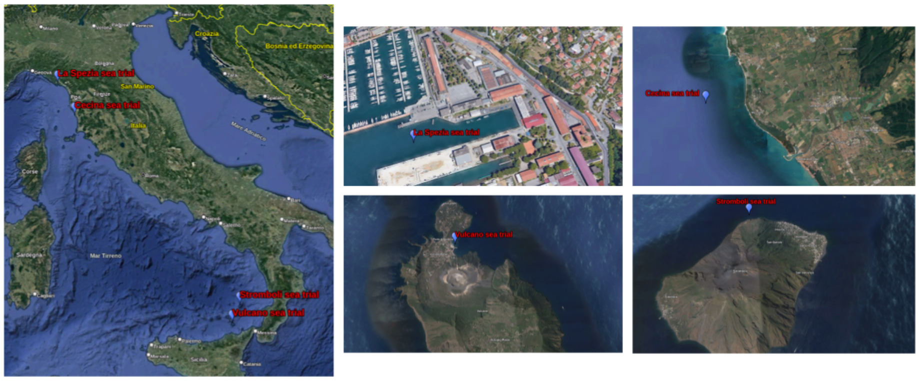


Figure 2.2: Locations of the sea trials, where the data employed to compute the experimental results presented in this thesis have been acquired. In particular, FeelHippo AUV has been employed in experimental campaigns in La Spezia (Italy), in Cecina, Livorno (Italy), in Vulcano Island, Messina (Italy), and in Stromboli Island, Messina (Italy).

the offset caused by the unknown disturbances.
A set of novel UKF-based sensor fusion strategies for autonomous underwater navigation is thus shown and compared. Increasing the navigation performance is a complex but essential target to be achieved because it guarantees that the AUV can correctly perform its mission and accomplish its tasks. The proposed strategies have been adapted to be inserted in an UKF-based framework. To the author's best knowledge, the fusion of inertial, acoustic, and optical data in an UKF algorithm and the application of the presented sensor fusion strategies to AUV navigation is novel.
Finally, to overcome the limitations introduced by the Kalman filter strategies, which condense all the history into the last estimation, a sensor fusion MAP algorithm has been developed. Due to the complexity of retrieving navigation information in the underwater environment, a sensor fusion approach has been used. The performance and robustness of the visual SLAM algorithm heavily rely on the quality of the images and salient features. Consequently, the visual SLAM system has been fused with other sensing algorithms, such as the DVL. As shown previously, very few works still exist on underwater SLAM fusing data from a monocular camera and a DVL. Despite that, fusing an optical and an acoustic sensor in a MAP-based framework can take advantages from both sensors, which have an excellent complement to each other. This developed

solution can be employed to locate the vehicle and map the seabed at the same time in a unified framework.
In conclusion, the main contributions of this work are summarized in the following:

- The development of an initialization strategy to compensate orientation offset caused by unknown magnetic disturbances. The proposed algorithm, which relies on a LG-EKF approach, has been validated during several experimental campaigns to evaluate its capabilities and robustness.

- The adaptation of several Kalman filtering sensor fusion algorithms to be combined with an UKF framework and the testing with previously acquired inertial, optical and acoustic data.

- A systematic and complete comparison of the proposed UKF-based solutions for underwater navigation. Centralized and decentralized strategies have been evaluated to determine the best approach regarding localization accuracy, robustness against outliers, and measurement reduction and consistency.

- The development of a sensor fusion algorithm relying on a MAP technique to fuse information coming from a monocular camera and a DVL and to consider the landmark points in the navigation framework. After validating the solution through realistic simulations, an experimental campaign at sea was conducted.

- An automatic tool to post-process the point cloud estimated through the SLAM algorithm. The seabed point cloud can be processed to obtain a 3D textured reconstruction, a geo-referenced depth map, and an error map of the performed reconstruction if a bathymetric ground truth is available.

Chapter 3

Preliminaries and notation

THIS chapter covers the notation employed in the rest of the work and provides a complete review of the fundamental theoretical and mathematical concepts used throughout this thesis.

3.1 Preliminaries on orientation initialization

3.1.1 Introduction to Lie group theory

A Lie group G is a manifold whose elements satisfy the group axioms. On one hand, the manifold is differentiable (or smooth), as it is a topological space that locally resembles linear space. On the other hand, considering the elements $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in G$ and defining the composition operation $\circ$, the group satisfies the following axioms:

- closure under $\circ$, as the composition of elements of the manifold remains on the manifold
$$\mathcal{X} \circ \mathcal{Y} \in G \tag{3.1}$$
- existence of the identity element $\mathcal{E}$
$$\mathcal{E} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{E} = \mathcal{X} \tag{3.2}$$
- existence of the inverse element in the manifold
$$\mathcal{X}^{-1} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{X}^{-1} = \mathcal{E} \tag{3.3}$$

- associativity properties

$$(\mathcal{X} \circ \mathcal{Y}) \circ \mathcal{Z} = \mathcal{X} \circ (\mathcal{Y} \circ \mathcal{Z}) \tag{3.4}$$

In summary, a manifold is a mathematical object that can be locally approximated to a piece of $\mathbb{R}^n$ and G is smooth if an appropriate dimension tangent space can be defined at each point.
The tangent space at the identity is called Lie algebra of G and noted as

$$\mathfrak{g} \triangleq T_{\mathcal{E}} G. \tag{3.5}$$

Every Lie group has and associated Lie algebra and it is possible to relate the Lie group with the Lie algebra through the following properties:

- the Lie algebra $\mathfrak{g}$ is a vector space and its elements can be identified with vectors in $\mathbb{R}^m$, whose dimension m is the number of DOFs of G;
- the exponential map $\exp_G : \mathfrak{g} \to G$ exactly converts elements of the Lie algebra into elements of the group and the logarithmic map $\log_G : G \to \mathfrak{g}$ perform the inverse operation;
- vectors of the tangent space at $\mathcal{X}$ can be transformed to the tangent space at the identity $\mathcal{E}$ through a linear transform, which is called adjoint transform.

The real matrix Lie group $G \subset \mathrm{GL}(n, \mathbb{R}^n)$ is a particular Lie group where each element of the group is a square $n \times n$ invertible matrix with real elements, the group operations are the matrix multiplication and inversion and the identity element is $\mathcal{I}^{n\times n}$.
A Lie algebra $\mathfrak{g}$ is an open neighborhood of $0^{n\times n}$ in the tangent space of G at the identity $\mathcal{I}^{n\times n}$. The matrix exponential map $\exp_G : \mathfrak{g} \to G$ and its inverse map, as the matrix logarithm map $\log_G : G \to \mathfrak{g}$, establish a local diffeomorphism between Lie groups and Lie algebra.
As the Lie algebra $\mathfrak{g}$ associated to a p-dimensional matrix Lie group $G \subset \mathbb{R}^{n\times n}$ is a vector space, a linear isomorphism between $\mathfrak{g}$ and $\mathbb{R}^p$ is given by

$$\begin{array}{l} [\cdot]_G^{\vee} : \mathfrak{g} \to \mathbb{R}^p \\ [\cdot]_G^{\wedge} : \mathbb{R}^p \to \mathfrak{g} \end{array}. \tag{3.6}$$

The adjoint representation of G on $\mathbb{R}^p$

$$\mathrm{Ad}_G : \mathrm{Ad}_G(X)x = \left[X[x]_G^{\wedge} X^{-1}\right]_G^{\vee} \tag{3.7}$$

and the adjoint representation of $\mathbb{R}^p$ on $\mathbb{R}^p$

$$\mathrm{ad}_G : \mathrm{ad}_G(x)y = [[x]^\wedge_G [y]^\wedge_G - [y]^\wedge_G [x]^\wedge_G]^\vee_G \tag{3.8}$$

where $x, y \in \mathbb{R}^p$ and $X \in G$, allow to assert that the Lie groups are not necessarily commutative. Finally, it is necessary to introduce the left-Jacobian matrix for Lie groups, which can be defined as

$$J^l_G\left([X]^\vee_G\right) = \begin{bmatrix} \left[\frac{\partial g}{\partial x_1} g^{-1}\right]^\vee_G & \dots & \left[\frac{\partial g}{\partial x_p} g^{-1}\right]^\vee_G \end{bmatrix}. \tag{3.9}$$

The SO(2) group is the group of special orthogonal matrices in the plane, i.e. rotation matrices. The Lie group and Lie algebra can be respectively defined as

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3.10}$$

$$[\theta]^\wedge_{\mathrm{SO}(2)} = \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix}. \tag{3.11}$$

Inversion can be achieved by transposition, i.e. $R_\theta^{-1} = R_\theta^\top$, and the product $R_\alpha \circ R_\beta = R_\alpha R_\beta$ is also member of the Lie group SO(2).
The exponential and logarithmic maps can be easily defined as

$$R_\theta = \exp\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3.12}$$

$$\theta = \log R_\theta = \mathrm{atan2}(r_{21}, r_{11}). \tag{3.13}$$

Considering the definitions and the equations above introduced, the following derivative blocks, which are useful for the LG-EKF design, are trivial.

$$\mathrm{Ad}(R_\theta) = 1 \tag{3.14}$$

$$J^{R_\theta^{-1}}_{R_\theta} = -1 \tag{3.15}$$

$$J^{R_\alpha R_\beta}_{R_\alpha} = J^{R_\alpha R_\beta}_{R_\beta} = 1 \tag{3.16}$$

$$J^{R_\theta \boldsymbol{v}}_{R_\theta} = R_\theta [1]^\wedge_{\mathrm{SO}(2)} \boldsymbol{v} = R_\theta \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \boldsymbol{v} \tag{3.17}$$

Eq. 3.17 represents the Jacobian of a rotation matrix action on a vector $\boldsymbol{v} \in \mathbb{R}^2$

and will be useful in the following. Further information regarding the Lie groups theory and its application to the SO(2) group can be found in [55].

3.1.2 Extended Kalman Filter on Lie groups

The LG-EKF is a generalization of the standard EKF [17], where the estimated state and the measurements have the structure of a Lie group [56], [57]. As in the classical Kalman filtering strategies, a system model has to be defined for the prediction step. Considering a p-dimensional matrix Lie group G, the system dynamics can be defined as

$$X_{k+1} = f(X_k, \boldsymbol{u}_k, \boldsymbol{n}_k) = X_k \exp_G\left(\left[\hat{\Omega}_k + \boldsymbol{n}_k\right]_G^\wedge\right) \tag{3.18}$$

where $X_k \in G$ is the system state at time k, $\boldsymbol{u}_k \in \mathbb{R}^w$ is the control input at time k, $\boldsymbol{n}_k \sim \mathcal{N}_{\mathbb{R}^p}\left(\mathbf{0}^{p\times 1}, Q_k\right)$ is a white Gaussian noise and $\hat{\Omega}_k = \Omega(X_k, u_k) : G \times \mathbb{R}^w \to \mathbb{R}^p$ is the non-linear system state equation which describes how the model operates on the state and control input to compute the displacement.
The measurement model can be provided by the following equation, where the available measurements are considered as part of a q-dimensional matrix Lie group G'.

$$z_k = h_k(X_k) \exp_{G'}\left([\boldsymbol{m}_k]_{G'}^\wedge\right) \tag{3.19}$$

where $z_k \in G'$ is the measurement at time k, $h_k(\cdot) : G \to G'$ defines the measurement map and $\boldsymbol{m}_k \sim \mathcal{N}_{\mathbb{R}^q}\left(\mathbf{0}^{q\times 1}, R_k\right)$ is a white Gaussian noise. It is worth noting that a different group G' is used since the system state and measurements might belong to different groups.
Assuming that the posterior state distribution after the $(k-1)$-th measurement is a concentrated Gaussian variable $X_{k-1|k-1} \sim \mathcal{N}_G\left(\mu_{k-1|k-1}, P_{k-1|k-1}\right)$, the prediction step of the LG-EKF to propagate the state mean and covariance is defined as

$$\mu_{k|k-1} = \mu_{k-1|k-1} \exp_G\left([\hat{\Omega}_{k-1}]_G^\wedge\right) \tag{3.20}$$

$$P_{k|k-1} = \mathcal{F}_{k-1} P_{k-1|k-1} \mathcal{F}_{k-1}^\top + \Phi_G(\hat{\Omega}_{k-1}) Q_{k-1} \Phi_G(\hat{\Omega}_{k-1})^\top \tag{3.21}$$

where

$$\mathcal{F}_{k-1} = \mathrm{Ad}_G\left(\exp_G\left([-\hat{\Omega}_{k-1}]_G^\wedge\right)\right) + \Phi(\hat{\Omega}_{k-1}) \mathcal{C}_{k-1} \tag{3.22}$$

$$\mathcal{C}_{k-1} = \frac{\partial}{\partial \boldsymbol{\epsilon}} \Omega\left(\mu_{k-1|k-1} \exp_G\left([\boldsymbol{\epsilon}]_G^\wedge\right)\right)\Big|_{\boldsymbol{\epsilon}=\mathbf{0}} \tag{3.23}$$

$$\Phi_G(\boldsymbol{\nu}) = \sum_{m=0}^{\infty} \frac{(-1)^m}{(m+1)!} \mathrm{ad}_G(\boldsymbol{\nu})^m = J_G^l(-\boldsymbol{\nu}). \tag{3.24}$$

It is necessary to notice that the parameter $\boldsymbol{\epsilon} \in \mathbb{R}^p$ represents a sort of Lie algebraic error, which can be approximated with a classical Euclidean Gaussian distribution, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}_{\mathbb{R}^p}\left(\mathbf{0}^{p\times 1}, P_k\right)$.
To proceed with the update step, it is necessary to compute the Kalman gain with the following equation:

$$K_k = P_{k|k-1}\mathcal{H}_k^\top \left(\mathcal{H}_k P_{k|k-1}\mathcal{H}_k^\top + R_k\right)^{-1} \tag{3.25}$$

where the measurement matrix $\mathcal{H}_k$ can be defined as

$$\mathcal{H}_k = \frac{\partial}{\partial \boldsymbol{\epsilon}} \left[\log_{G'}\left(h_k(\mu_{k|k-1})^{-1} h_k\left(\mu_{k|k-1}\exp_G\left([\boldsymbol{\epsilon}]_G^\wedge\right)\right)\right)\right]_G^\vee \Bigg|_{\boldsymbol{\epsilon}=\mathbf{0}}. \tag{3.26}$$

Subsequently, the innovation vector multiplied by the Kalman gain can be computed as

$$\boldsymbol{\nu}_k = K_k \left[\log_{G'}\left(h_k(\mu_{k|k-1})^{-1} z_k\right)\right]_{G'}^\vee. \tag{3.27}$$

Finally, the update of the system state and its covariance matrix can be performed as follows:

$$\mu_{k|k} = \mu_{k|k-1}\exp_G\left([\boldsymbol{\nu}_k]_G^\wedge\right) \tag{3.28}$$

$$P_{k|k} = \Phi_G(\boldsymbol{\nu}_k)\left(\mathcal{I}^{p\times p} - K_k\mathcal{H}_k\right) P_{k|k-1}\Phi_G(\boldsymbol{\nu}_k)^\top. \tag{3.29}$$

The LG-EKF equations are summarized in Algorithm 1.

3.2 Preliminaries on UKF-based fusion strategies

3.2.1 Kinematic and dynamic modeling of the AUV

Firstly, the mathematical notation used in this thesis to describe the motion of a general rigid body freely moving within a fluid is discussed in this section. Further information can be found in [58]. The main assumptions, which form the basis of the following discussion, consider any frame located on the Earth's surface as inertial (the rotation of the Earth is therefore neglected) and any vehicle involved as a rigid body. The Society of Naval Architects and Marine Engineers (SNAME) notation, commonly adopted for marine vehicles, is used

Algorithm 1 LG-EKF algorithm

```
Function LGEKF():
    /* prediction step                                          */
    Step Prediction():
        Input  : $\mu_{k-1|k-1}, P_{k-1|k-1}, \hat{\Omega}_{k-1}$
        Output: $\mu_{k|k-1}, P_{k|k-1}$
        $\mu_{k|k-1} = \mu_{k-1|k-1} \exp_G \left( [\hat{\Omega}_{k-1}]_G^\wedge \right)$;
        $P_{k|k-1} = \mathcal{F}_{k-1} P_{k-1|k-1} \mathcal{F}_{k-1}^\top + \Phi_G(\hat{\Omega}_{k-1}) Q_{k-1} \Phi_G(\hat{\Omega}_{k-1})^\top$;
    end
    /* correction step                                          */
    Step Correction():
        Input  : $\mu_{k|k-1}, P_{k|k-1}, h_k(\cdot)$
        Output: $\mu_{k|k}, P_{k|k}$
        $S_k = \mathcal{H}_k P_{k|k-1} \mathcal{H}_k^\top + R_k$;
        $K_k = P_{k|k-1} \mathcal{H}_k^\top S_k^{-1}$;
        $\boldsymbol{\nu}_k = K_k \left[ \log_{G'} \left( h_k(\mu_{k|k-1})^{-1} z_k \right) \right]_{G'}^\vee$;
        $\mu_{k|k} = \mu_{k|k-1} \exp_G \left( [\boldsymbol{\nu}_k]_G^\wedge \right)$;
        $P_{k|k} = \Phi_G(\boldsymbol{\nu}_k) \left( \mathcal{I}^{p \times p} - K_k \mathcal{H}_k \right) P_{k|k-1} \Phi_G(\boldsymbol{\nu}_k)^\top$;
    end
end
```

throughout the thesis.

The pose of the AUV (in terms of position and attitude) is retrieved with respect to two reference frames. The first one is a local Earth-fixed reference frame, as theNorth, East, Down (NED) frame $\{O^N x^N y^N z^N\}$, whose axes point, respectively, North, East, and Down. The second one, namely the body frame, is usually centered on the Center of Gravity (CG) of the vehicle, with the forward motion direction represented by the x-axis (surge) and the z-axis (heave) pointing down. Lastly, the y-axis (sway) completes a right-handed reference frame $\{O^b x^b y^b z^b\}$.

The AUV pose with respect to the NED frame is represented with the vector $\boldsymbol{\eta} = \begin{bmatrix} {}^N\boldsymbol{\eta}_1^\top & \boldsymbol{\eta}_2^\top \end{bmatrix}^\top$, where ${}^N\boldsymbol{\eta}_1$ is the position of the CG of the vehicle with respect to the NED frame and $\boldsymbol{\eta}_2$ is its orientation, as the rotation angles between the body and the NED frames. In particular, a triplet of Euler angles expressed with respect to the fixed NED frame, namely Roll, Pitch and Yaw (RPY), is used to define $\boldsymbol{\eta}_2$, where the roll, pitch, and yaw angles are respectively denoted with ϕ, θ, and ψ. Moreover, the vector ${}^b\boldsymbol{\nu} = \begin{bmatrix} {}^b\boldsymbol{\nu}_1^\top & {}^b\boldsymbol{\nu}_2^\top \end{bmatrix}^\top$ contains the linear and angular velocities of the AUV expressed in the body-fixed reference frame. In particular, u, v, and w are the linear velocities along the axes of the body frame (surge, sway, and heave), and p, q, and r are the angular counterparts of the axes mentioned above. The AUV kinematic variables can be summarized as:

$$\begin{aligned} \boldsymbol{\eta} &= \begin{bmatrix} {}^N\boldsymbol{\eta}_1^\top & \boldsymbol{\eta}_2^\top \end{bmatrix}^\top, \\ {}^b\boldsymbol{\nu} &= \begin{bmatrix} {}^b\boldsymbol{\nu}_1^\top & {}^b\boldsymbol{\nu}_2^\top \end{bmatrix}^\top \end{aligned} \tag{3.30}$$

where

$$\begin{aligned} {}^N\boldsymbol{\eta}_1 &= [x \quad y \quad z]^\top, & \boldsymbol{\eta}_2 &= [\phi \quad \theta \quad \psi]^\top, \\ {}^b\boldsymbol{\nu}_1 &= [u \quad v \quad w]^\top, & {}^b\boldsymbol{\nu}_2 &= [p \quad q \quad r]^\top. \end{aligned} \tag{3.31}$$

The complete kinematic model of the vehicle is reported in Eq. 3.32.

$$\begin{bmatrix} {}^N\dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{bmatrix} = \begin{bmatrix} R_b^N(\boldsymbol{\eta}_2) & 0_{3\times3} \\ 0_{3\times3} & T_b^N(\boldsymbol{\eta}_2) \end{bmatrix} \begin{bmatrix} {}^b\boldsymbol{\nu}_1 \\ {}^b\boldsymbol{\nu}_2 \end{bmatrix} \tag{3.32}$$

where $R_b^N(\boldsymbol{\eta}_2)$ represents the rotation matrix between the body and the fixed reference system (see Eq. 3.33) and $T_b^N(\boldsymbol{\eta}_2)$ is the Euler matrix (see Eq. 3.34).

$$R_b^N(\boldsymbol{\eta}_2) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi c_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{3.33}$$

$$T_b^N(\boldsymbol{\eta}_2) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \tag{3.34}$$

It can be noted that $T_b^N(\boldsymbol{\eta}_2)$ is not defined for $\theta = \pm 90°$; however, the majority of the AUVs does not operate close to this singularity. The compact form of Eq. 3.32 is reported in Eq. 3.35, where the definition of $J(\boldsymbol{\eta})$ is trivial.

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{\nu} \tag{3.35}$$

According to [58], the AUV complete dynamic model is:

$$M\dot{\boldsymbol{\nu}} + C(\boldsymbol{\nu})\boldsymbol{\nu} + D(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}_{\boldsymbol{\eta}}(\boldsymbol{\eta}) = \boldsymbol{\tau}(\boldsymbol{\nu}, \mathbf{u}) \tag{3.36}$$

where M is the mass and inertia matrix, $C(\boldsymbol{\nu})$ is the Coriolis and centripetal matrix, $D(\boldsymbol{\nu})$ is the damping matrix, the vector $\boldsymbol{g}_{\boldsymbol{\eta}}(\boldsymbol{\eta})$ contains the buoyancy and gravity effects and $\boldsymbol{\tau}(\boldsymbol{\nu}, \mathbf{u})$ takes into account the thrust action on the AUV, which depends on the vehicle speed $\boldsymbol{\nu}$ and the rotational speed of each motor $\mathbf{u} \in \mathbb{R}^m$, where m is the number of motors.
For the sake of clarity, each term in Eq. 3.36 is briefly described.
The vector of forces and moments acting on the vehicle $\boldsymbol{\tau}(\boldsymbol{\nu}, \mathbf{u})$ (usually referred in the body-fixed frame) can be represented as

$$\boldsymbol{\tau}(\boldsymbol{\nu}, \mathbf{u}) = \begin{bmatrix} {}^b\boldsymbol{\tau}_1^\top & {}^b\boldsymbol{\tau}_2^\top \end{bmatrix}^\top \tag{3.37}$$

where ${}^b\boldsymbol{\tau}_1 = \begin{bmatrix} X & Y & Z \end{bmatrix}^\top$ and ${}^b\boldsymbol{\tau}_2 = \begin{bmatrix} K & M & N \end{bmatrix}^\top$. The generalized forces $\boldsymbol{\tau}$, the vector $\boldsymbol{T} \in \mathbb{R}^m$ that collects the thrusts exerted by the m motors of the vehicle and the rotational speed of the motors $\boldsymbol{u} \in \mathbb{R}^m$ are linked using the following relation:

$$\boldsymbol{\tau}(\boldsymbol{\nu}, \mathbf{u}) = B\boldsymbol{T}(\boldsymbol{\nu}, \mathbf{u}), \tag{3.38}$$

where $B \in \mathbb{R}^{6\times m}$ is a constant matrix that depends upon the thruster poses with respect to the CG. B can be defined as:

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \tag{3.39}$$

with

$$\begin{aligned} B_1 &= \begin{bmatrix} \ldots & {}^b\boldsymbol{n}_{m,i} & \ldots \end{bmatrix} \\ B_2 &= \begin{bmatrix} \ldots & \left({}^b\boldsymbol{P}_{m,i} \times {}^b\boldsymbol{n}_{m,i}\right) & \ldots \end{bmatrix} \end{aligned} \tag{3.40}$$

where ${}^b\boldsymbol{n}_{m,i}$ is the axis of the i-th motor expressed in the body frame $\{O^b x^b y^b z^b\}$ and ${}^b\boldsymbol{P}_{m,i}$ is the position of the i-th motor with respect to the CG expressed in the body frame.
The mass and inertia matrix M is constituted of two terms:

$$M = M_{RB} + M_A, \tag{3.41}$$

where M_{RB} depends upon vehicle geometry and material and M_A is the added mass matrix. Assuming that the body frame is centered on the CG of the vehicle and that the body-fixed frame axes coincide with the principal axes of inertia, it is possible to assert that M_{RB} becomes:

$$M_{RB} = \begin{bmatrix} \mathrm{diag}\{ m, \quad m, \quad m \} & 0_{3\times 3} \\ 0_{3\times 3} & \mathrm{diag}\{ I_x, \quad I_y, \quad I_z \} \end{bmatrix}, \tag{3.42}$$

where m is the mass of the vehicle and I_x, I_y and I_z are the principal moments of inertia (respectively about surge, sway and heave axes).
According to [58], added mass can be considered as forces and moments induced by the motion of the vessel body proportional to its acceleration. In fact, any motion of the AUV will produce motion in the surrounding fluid; hence, the fluid motion possesses kinetic energy that it would lack otherwise. This phenomenon is experienced as an additional inertia when the vehicle, and therefore the surrounding fluid, accelerates. The general expression of M_A presents 36 parameters but, under the assumptions stated in [59], the added mass matrix becomes:

$$M_A = -\mathrm{diag}\left\{\; X_{\dot{u}}, \quad Y_{\dot{v}}, \quad Z_{\dot{w}}, \quad K_{\dot{p}}, \quad M_{\dot{q}}, \quad N_{\dot{r}} \;\right\} \tag{3.43}$$

where the SNAME notation is employed for forces, torques and accelerations. To clarify, each element $A_{\dot{b}} = \frac{\partial A}{\partial \dot{b}}$ in Eq. 3.43 is the coefficient to compute the force (torque) $-A_{\dot{b}}\dot{b}$ along (about) the body a-axis caused by an acceleration $\dot{b}$ that, according to the SNAME notation, is the acceleration along (about) the body a-axis.
The matrix $C(\boldsymbol{\nu})$ is constituted of two terms:

$$C(\boldsymbol{\nu}) = C_{RB}(\boldsymbol{\nu}) + C_A(\boldsymbol{\nu}), \tag{3.44}$$

where $C_{RB}(\boldsymbol{\nu})$ is due to centripetal and Coriolis effects and $C_A(\boldsymbol{\nu})$ takes into account the motion of the vehicle within the fluid. Under the same assumptions made previously for the computation of the mass and inertia matrix, it can be shown that $C_{RB}(\boldsymbol{\nu})$ can be represented as:

$$C_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & -mr & mq & 0 & 0 & 0 \\ mr & 0 & -mp & 0 & 0 & 0 \\ -mq & mp & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_z r & -I_y q \\ 0 & 0 & 0 & -I_z r & 0 & I_x p \\ 0 & 0 & 0 & I_y q & -I_x p & 0 \end{bmatrix}. \tag{3.45}$$

Moreover, $C_A(\boldsymbol{\nu})$ can be expressed as follows:

$$C_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix}. \tag{3.46}$$

Matrix $D(\boldsymbol{\nu})$ models the dissipative effects due to the motion within the fluid, which are mainly caused by nonlinear skin friction due to turbulent boundary layers and by viscous damping force due to vortex shedding. For an underwater vehicle moving at low speed, $D(\boldsymbol{\nu})$ is often assumed as diagonal with quadratic damping terms only, which dominate over linear terms. Coupling effects and terms higher than the second order with respect to the body-fixed velocity can be neglected. Consequently, $D(\boldsymbol{\nu})$ can be approximated with

$$D(\boldsymbol{\nu}) = -\text{diag}\left\{\; X_{u|u|}|u|, \quad Y_{v|v|}|v|, \quad Z_{w|w|}|w|, \quad K_{p|p|}|p|, \quad M_{q|q|}|q|, \quad N_{r|r|}|r| \;\right\}. \tag{3.47}$$

Gravitational and buoyancy effects acting on the vehicle can be computed separately and the combined into the vector $\boldsymbol{g}_{\boldsymbol{\eta}}(\boldsymbol{\eta})$. In particular, the gravitational force, applied to the CG and expressed in the NED reference frame, can be computed as

$${}^N\mathbf{W} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \tag{3.48}$$

where m is the dry mass of the vehicle and $g = 9.806$ m/s^2 is the gravitational acceleration. Buoyancy, on the other side, is applied to the Center of Buoyancy (CB) of the vehicle, that usually does not coincide with the CG, and is equal to

$${}^N\mathbf{B} = -\begin{bmatrix} 0 \\ 0 \\ \rho g V \end{bmatrix}, \tag{3.49}$$

where ρ is the water density and V denotes the total volume of the vehicle. Reporting the quantities in the body frame

$${}^b\mathbf{W} = (R_b^N)^\top {}^N\mathbf{W} \tag{3.50}$$

$$
{}^{b}\mathbf{B} = (R_b^N)^\top {}^{N}\mathbf{B}, \tag{3.51}
$$

the vector $\boldsymbol{g}_{\boldsymbol{\eta}}(\boldsymbol{\eta})$ can be expressed as follows:

$$
\boldsymbol{g}_{\boldsymbol{\eta}}(\boldsymbol{\eta}) = \begin{bmatrix} {}^{b}\mathbf{W} + {}^{b}\mathbf{B} \\ {}^{b}\mathbf{r}_b \times {}^{b}\mathbf{B} \end{bmatrix}, \tag{3.52}
$$

where ${}^{b}\mathbf{r}_b$ is the position of the CB with respect to the body frame centered in the CG and × indicates the cross product.

3.2.2 Unscented Kalman Filter

Given a generic dyanamical system, it is possible to define a set of variables (usually representing physical quantities) which completely describe its state. Generally speaking, the evolution of a system is described by the following nonlinear time-varying equation:

$$
\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\right), \tag{3.53}
$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state vector, $\boldsymbol{f}(\cdot)$ is a generic nonlinear, time-varying function, $\boldsymbol{u}(t) \in \mathbb{R}^m$ is the vector of the controlled inputs and $\boldsymbol{w}(t) \in \mathbb{R}^n$ is the process noise used to describe modeling uncertainties. The physical phenomena describing the behavior of the state variables are not completely known and $\boldsymbol{f}(\cdot)$ only represents a model for the evolution of the system. As a consequence, the process noise $\boldsymbol{w}(t)$ is employed.

Considering that the direct knowledge of the state variables might be unavailable, it is usually possible to dispose of a set of measurements, obtained from sensors, which are related to the state variables themselves. A generic measurement equation can be described as:

$$
\boldsymbol{y}(t) = \boldsymbol{h}\left(t, \boldsymbol{x}(t), \boldsymbol{v}(t)\right), \tag{3.54}
$$

where $\boldsymbol{y}(t) \in \mathbb{R}^p$ is the measurement vector, $\boldsymbol{h}(\cdot)$ is a generic nonlinear, time-varying function and $\boldsymbol{v}(t) \in \mathbb{R}^p$ is the measurement noise used to model sensor accuracy. As stated before for the system evolution model, the sensor models are not completely accurate and, thus, the measurement noise $\boldsymbol{v}(t)$ is employed. Together, Eq. 3.53 and Eq. 3.54 are usually referred as state-space representation of the system.

The previous introduced treatment is verified for continuous system (where the time $t \in \mathbb{R}$), so it is necessary to introduce a discrete-time representation of Eq.

3.53 and Eq. 3.54. Given a uniform sampling of the time-continuous variables and functions, the state-space representation of the system can be rewritten as follows:

$$\begin{cases} \boldsymbol{x}_k = \boldsymbol{f}_{k-1}\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}\right) \\ \boldsymbol{y}_k = \boldsymbol{h}_k\left(\boldsymbol{x}_k, \boldsymbol{v}_k\right) \end{cases}, \tag{3.55}$$

where k indicates the iteration number.

Algorithm 2 UKF algorithm

```
Function UKF():
    /* prediction step                                                    */
    Step Prediction():
        Input  : $\hat{\boldsymbol{x}}_{k-1|k-1}, P_{k-1|k-1}, \boldsymbol{f}_{k-1}(\cdot)$
        Output: $\hat{\boldsymbol{x}}_{k|k-1}, P_{k|k-1}$
        $\left(\hat{\boldsymbol{x}}_{k|k-1}, \bar{P}_{k|k-1}\right) = UT\left(\hat{\boldsymbol{x}}_{k-1|k-1}, P_{k-1|k-1}, \boldsymbol{f}_{k-1}(\cdot)\right)$;
        $P_{k|k-1} = \bar{P}_{k|k-1} + Q_{k-1}$;
    end
    /* correction step                                                    */
    Step Correction():
        Input  : $\hat{\boldsymbol{x}}_{k|k-1}, P_{k|k-1}, \boldsymbol{h}_k(\cdot)$
        Output: $\hat{\boldsymbol{x}}_{k|k}, P_{k|k}$
        $\left(\hat{\boldsymbol{y}}_{k|k-1}, \bar{S}_k, P_k^{xy}\right) = UT\left(\hat{\boldsymbol{x}}_{k|k-1}, P_{k|k-1}, \boldsymbol{h}_k(\cdot)\right)$;
        $S_k = \bar{S}_k + R_k$;
        $L_k = P_k^{xy} S_k^{-1}$;
        $\boldsymbol{e}_k = \boldsymbol{y}_k - \boldsymbol{y}_{k|k-1}$;
        $\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + L_k \boldsymbol{e}_k$;
        $P_{k|k} = P_{k|k-1} - L_k S_k L_k^\top$;
    end
end
```

When the system state is not available, suitable observers (or estimators) need to be employed to obtain an accurate estimation of the system behavior. The majority of state estimators for dynamical system are based on the Bayesian statistics, such as on the KF [16] or its variants applied to nonlinear systems, as the EKF [17] or the UKF [19], [60]. The KF or its variants rely on a recursive observer based on the alternation of two steps:

- prediction, where the behavior of the system at the subsequent iteration is predicted exploiting the current estimate and the model;
- update (or correction), where the predicted state is corrected using the available measurements to produce an improved state estimate.

During the Ph.D. period, the UKF-based navigation filter position estimation of an AUV, already developed and validated by the UNIFI DIEF, has been further enhanced to manage speed measurements from several available sensors and to fuse them according to appropriate strategies. Therefore, its structure and main features are described here below.
The UKF is based on the Unscented Transform (UT), a deterministic sampling technique which allows the computation of the mean and the covariance matrix of a Random Variable (RV) which undergoes a generic nonlinear transformation by propagating a minimum set of its samples and exploiting the knowledge of the mean and of the covariance of the starting variable. As reported in [60], [19], the variable moments computed using the UT are accurate at least to the second order of the Taylor series expansion of the considered nonlinear function.
The UT, in case of nonlinear state evolution and measurement functions, is used for both the prediction and the correction steps. Assuming additive noise with zero-mean, white Gaussian distribution, an implementation of the UKF algorithm is reported in Algorithm 2, where $\hat{\boldsymbol{x}}_{j|k}$ and $P_{j|k}$ are respectively the state estimate and its covariance at iteration j given the information up to the k-th iteration, Q is the process noise, R is the measurement noise and $(\hat{\boldsymbol{x}}_{0|0}, P_{0|0})$ is the initial guess for the state and the state covariance.

3.2.3 Sensor modeling for UKF-based strategies

A brief treatment, from a Kalman filtering point of view, on the mathematical modeling of sensors and devices that are usually used in marine robotics for autonomous navigation and mapping is presented.

Global Positioning System. Although the GPS signal cannot be employed in underwater scenarios [61], it is usually mounted on almost all AUVs for initialization and/or periodic position resets. A GPS measurement can be modeled as in Eq. 3.56.

$$^{N}\boldsymbol{P}_{GPS} = \boldsymbol{f}_{GPS}(\boldsymbol{P}_{GPS} + \tilde{\boldsymbol{\epsilon}}_{GPS}, \boldsymbol{O}) = {}^{N}\boldsymbol{\eta}_1 + \boldsymbol{\epsilon}_{GPS}, \tag{3.56}$$

where $^{N}\boldsymbol{P}_{GPS}$ is the position in the NED frame obtained after the use of $\boldsymbol{f}_{GPS}$, the function to convert geodetic coordinates to local NED ones, $\boldsymbol{P}_{GPS}$ is the position obtained with the GPS, $\tilde{\boldsymbol{\epsilon}}_{GPS}$ is the noise term added to model sensor accuracy, which is mapped into $\boldsymbol{\epsilon}_{GPS}$ after the NED conversion, and $\boldsymbol{O}$ are the geodetic coordinates of the point assumed as origin of the NED frame. It is worth noting that, after the conversion into NED coordinates, the resulting measurement noise $\boldsymbol{\epsilon}_{GPS}$ is supposed to remain additive, but it possess different spectral properties than those of the noise affecting raw data.

Depth Sensor. A DS uses pressure measurements in order to estimate depth. The stochastic representation of a pressure measurement can be obtained as:

$$p_{DS} = p + b_p + \epsilon_p, \tag{3.57}$$

where p_{DS} is the measured pressure, obtained as sum of the real pressure p, the sensor bias b_p and the measurement noise ϵ_p. The pressure measurement can be converted into a depth measurement thanks to the hydrostatic relation reported in Eq. 3.58.

$$p_{DS} - p_0 = \rho g^N d_{DS} \tag{3.58}$$

where p_0 is the pressure measured during the initialization phase, ρ is the water density and d_{DS} is the measured depth, which can be expressed as:

$${}^N d_{DS} = z + \epsilon_{DS}, \tag{3.59}$$

being z the real depth and ϵ_{DS} the noise. Note that the pressure bias b_p cancels in the subtraction in Eq. 3.58.

Doppler Velocity Log. A DVL is a device capable to measure, thanks to the Doppler effect, a 3D linear velocity of the vehicle in the body reference frame with respect to the sea bottom. As a consequence, a DVL measurement can be modeled as in the following equation:

$${}^b\boldsymbol{v}_{DVL} = {}^b\boldsymbol{\nu}_1 + \boldsymbol{b}_{DVL} + \boldsymbol{\epsilon}_{DVL}, \tag{3.60}$$

where the measured velocity ${}^b\boldsymbol{v}_{DVL}$ is the sum of the true body-fixed velocity of the sensor bias $\boldsymbol{b}_{DVL}$ and of the measurement noise $\boldsymbol{\epsilon}_{DVL}$.

Inertial Measurement Unit. Despite the classical definition of IMU indicates an instrument possessing accelerometers and gyroscopes only, it is not uncommon for IMU to integrate a triaxial magnetometer, used to measure the magnetic field surrounding the sensor. The following mathematical models have been adopted for the accelerometers, the gyroscopes and the magnetometers, where all the measured quantities are expressed in the body frame.

- For what concern the accelerometers,

$$ {}^{b}\boldsymbol{a}_{IMU} = (R_b^N)^\top \left(\ddot{\boldsymbol{\eta}}_1 - \boldsymbol{g}\right) + \boldsymbol{b}_a + \boldsymbol{\epsilon}_a, \tag{3.61} $$

  where $\boldsymbol{b}_a$ is the accelerometer bias and $\boldsymbol{\epsilon}_a$ is the measurement noise. When the accelerometers are properly calibrated the bias term is usually neglected.

- For what concern the gyroscopes,

$$ {}^{b}\boldsymbol{\omega}_{IMU} = {}^{b}\boldsymbol{\nu}_2 + \boldsymbol{b}_g + \boldsymbol{\epsilon}_g, \tag{3.62} $$

  where the sum of the true angular velocity ${}^{b}\boldsymbol{\nu}_2$, the bias $\boldsymbol{b}_g$ and the noise $\boldsymbol{\epsilon}_g$ produces the sensed quantity ${}^{b}\boldsymbol{\omega}_{IMU}$.

- For what concern the magnetometers,

$$ {}^{b}\boldsymbol{m} = W(R_b^N)^\top({}^{N}\boldsymbol{H}) + \boldsymbol{H}_m + \boldsymbol{\epsilon}_m, \tag{3.63} $$

  where ${}^{N}\boldsymbol{H}$ is the Earth's magnetic field expressed in the fixed frame, W and $\boldsymbol{H}_m$ are the disturbances caused by local magnetic interferences and $\boldsymbol{\epsilon}_m$ is the measurement noise.

Fiber Optic Gyroscope. A FOG is a device able to measure the angular rate thanks to the Sagnac effect. For what concerns a single-axis device, the employed mathematical model reported in Eq. 3.64 can be adopted.

$$ \omega_{FOG}^{m} = \omega_{FOG} + b_e + b_f + \epsilon_f, \tag{3.64} $$

where ω_{FOG}^{m} is the measured quantity, ω_{FOG} is the true value, b_e is the component of Earth's rotation sensed by the gyroscope, b_f is an additional bias term and ϵ_f is the measurement noise.

Forward Looking SONAR. A 2D FLS can provide a series of azimuth θ and range R measurements with a scalar value that represents the intensity of the returned echo (see Fig. 3.1). Considering a scene point $\begin{bmatrix} X & Y & Z \end{bmatrix}^\top$, the imaging projection model for an FLS can be obtained with the following equation [14]:

$$ \begin{bmatrix} {}^{m}R \\ {}^{m}\theta \end{bmatrix} = \begin{bmatrix} R \\ \theta \end{bmatrix} + \tilde{\boldsymbol{b}}_{FLS} + \tilde{\boldsymbol{\epsilon}}_{FLS} = \begin{bmatrix} \sqrt{X^2 + Y^2 + Z^2} \\ \tan^{-1}(\frac{Y}{X}) \end{bmatrix} + \tilde{\boldsymbol{b}}_{FLS} + \tilde{\boldsymbol{\epsilon}}_{FLS} \tag{3.65} $$

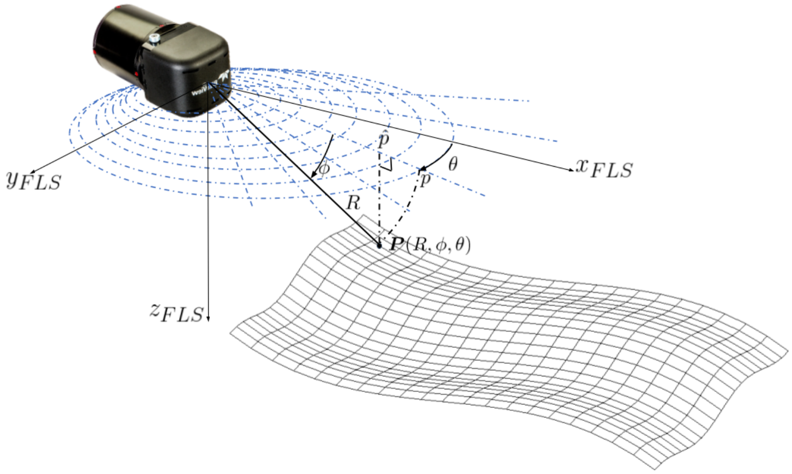


Figure 3.1: FLS imaging model.

where ${}^mR$ is the measured range and R is its real value, ${}^m\theta$ is the measured azimuth and θ is its real value and $\tilde{\boldsymbol{b}}_{FLS}$ and $\tilde{\boldsymbol{\epsilon}}_{FLS}$ are respectively the additional bias term and the measurement noise for both the range and the azimuth angle. Considering that the FLS is used to obtain linear speed estimations and that it is not easy to predict the final noise statistics, a model with a measured value that is the sum of a true one, a bias term and a noise can be adopted:

$$ {}^b\boldsymbol{v}_{FLS} = {}^b\boldsymbol{\nu}_1 + \boldsymbol{b}_{FLS} + \boldsymbol{\epsilon}_{FLS} \tag{3.66} $$

where the measured velocity ${}^b\boldsymbol{v}_{FLS}$ is assumed to be the sum of the true value ${}^b\boldsymbol{\nu}_1$, of the bias $\boldsymbol{b}_{FLS}$ and of the measurement noise $\boldsymbol{\epsilon}_{FLS}$. Necessarily, the noise statistics of the estimated speed will possess different spectral properties than those of the noise affecting range and elevation angle.

Monocular Camera. A monocular camera can provide a two-dimensional image, where to each pixel located in $\begin{bmatrix} x & y \end{bmatrix}^\top$ on the image plane it is possible to associate a scalar value that represents the lighting value (see Fig. 3.2). Considering a scene point $\begin{bmatrix} X & Y & Z \end{bmatrix}^\top$, the pinhole camera projection model with focal length f can be retrieved [62]:

$$ \begin{bmatrix} {}^mx \\ {}^my \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \tilde{\boldsymbol{b}}_{CAM} + \tilde{\boldsymbol{\epsilon}}_{CAM} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} + \tilde{\boldsymbol{b}}_{CAM} + \tilde{\boldsymbol{\epsilon}}_{CAM} \tag{3.67} $$

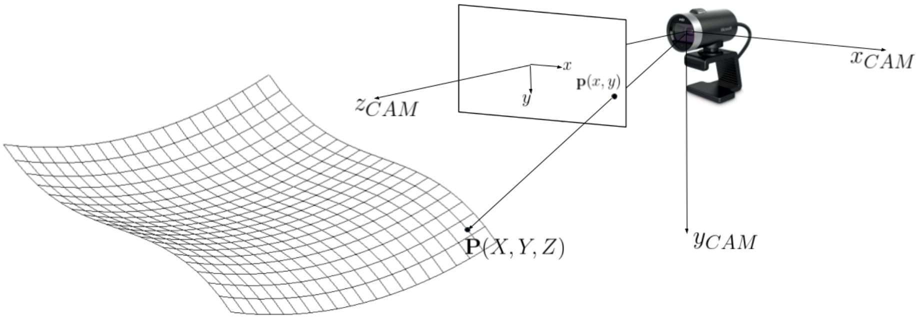


Figure 3.2: Camera projection model.

where $\begin{bmatrix} {}^m x & {}^m y \end{bmatrix}^\top$ is the measured pixel position and $\begin{bmatrix} x & y \end{bmatrix}^\top$ is its real value and $\tilde{\boldsymbol{b}}_{CAM}$ and $\tilde{\boldsymbol{\epsilon}}_{CAM}$ are respectively the additional bias term and the measurement noise for the x and y position. Considering that the camera is used to obtain linear speed estimations and that it is convenient to standardize the noise modeling, a model with a measured value that is the sum of a true one, a bias term and a noise can be employed:

$$ {}^b\boldsymbol{v}_{CAM} = {}^b\boldsymbol{\nu}_1 + \boldsymbol{b}_{CAM} + \boldsymbol{\epsilon}_{CAM} \tag{3.68}$$

where the measured velocity ${}^b\boldsymbol{v}_{CAM}$ is assumed to be the sum of the true value ${}^b\boldsymbol{\nu}_1$, of the bias $\boldsymbol{b}_{CAM}$ and of the measurement noise $\boldsymbol{\epsilon}_{CAM}$. Necessarily, the noise statistics of the estimated speed will possess different spectral properties than those of the noise affecting the pixel position.

3.3 Preliminaries on SLAM-based fusion strategies

3.3.1 Factor graph for SLAM algorithms

Factor graphs are graphical models [63] that can adapt well to modeling complex estimation problems, such as SLAM. Formally a factor graph is a bipartite graph $F = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ with two types of nodes, as the factors $\phi_i \in \mathcal{U}$ and the variables $x_j \in \mathcal{V}$. Referring to the navigation and mapping problems, while the variables represent the unknown random variables in the estimation problem (e.g., robot pose and landmark position), the factors represent probabilistic constraints on those variables derived from measurements or prior knowledge. Edges $e_{ij} \in \mathcal{E}$ are connections between factor nodes and variable nodes. Defining as X_i the set of variable nodes adjacent to a factor ϕ_i, the factorization of the global function $\phi(X)$, where X contains all the adjacency set X_i, is

$$\phi(X) = \prod_i \phi_i(X_i). \tag{3.69}$$

Summarizing, the independence relationships are encoded by the edges e_{ij} of the factor graph, where each factor ϕ_i is a function of only the variables X_i of the adjacency set.
Focusing on the navigation and mapping applications [35], it is necessary to analyze how tracking and SLAM problems can be defined and solved with factor graphs. In tracking, the MAP estimate of the trajectory X given the measurement Z can be obtained by maximizing the functional

$$p(X|Z) \propto p(x_1)l(z_1|x_1) \prod_{i>1} p(x_i|x_{i-1})l(z_i|x_i) \tag{3.70}$$

where i is the index at which the measurements z_i are observed. The posterior density $p(X|Z)$ has been factored into a set of unary likelihood factors $\phi(x_i) \propto l(z_i|x_i)$ and binary odometry factors $\phi(x_i, x_{i-1}) \propto p(x_i|x_{i-1})$. In tracking problems, only one type of unknown variable is considered, as the robot's state. When the knowledge of the environment is also important, it is necessary to introduce new variables related to surrounding landmarks, as in the SLAM problem. When landmarks are introduced in the estimation problem, new variables l_j and additional factors $\phi(x_i, l_j)$, one for each observation of a specific landmark l_j from a particular pose x_i should be inserted in the factor graph.

3.3.2 Maximum A Posteriori estimation

A navigation and mapping problem is a problem where the unknown state variables $X = \{x_1, x_2, ..., x_M\}$ constituted of poses and landmarks has to be determined given the measurements $Z = \{z_1, z_2, ..., z_N\}$. The MAP estimator maximizes the posterior density $p(X|Z)$ of the states X given the measurements Z:

$$X^{MAP} = \underset{X}{\operatorname{argmax}}\, p(X|Z) = \underset{X}{\operatorname{argmax}} \frac{p(Z|X)p(X)}{p(Z)} \tag{3.71}$$

The second expression can be found thanks to the Bayes law and define the posterior density as product of the measurement density $p(Z|X)$ and the prior density over the states $p(X)$, normalized by the factor $p(Z)$. Considering that the measurements Z are given, the normalization factor $p(Z)$ is irrelevant to the maximization problem. In addition, the likelihood $l(Z|X)$ of the states X given the measurements Z is proportional to the probability $p(Z|X)$. Thus, Eq. 3.71 becomes:

$$X^{MAP} = \underset{X}{\operatorname{argmax}}\, p(X)l(Z|X) = p(X)\prod_{i=1}^{N} l(z_i|X), \tag{3.72}$$

where an additive Gaussian noise is assumed in all measurement models, as reported in Eq. 3.73.

$$p(z_i|X) = \mathcal{N}(h_i(X), \Sigma_i) \propto \exp\left(-\frac{1}{2}\|h_i(X) - z_i\|^2_{\Sigma_i}\right) \tag{3.73}$$

where $h_i(X)$ is the measurement function, which maps the state estimate X into a predicted value $\hat{z}_i$ of the measurement z_i and Σ_i is the covariance matrix, which summarizes the uncertainty of the measurement model. It is worth noting that, in general, z_i is conditioned on a subset of the variables belonging to X. Summarizing, each factor associated to a measurement can be defined with three components, as the measurement itself, the prediction function and the noise model. By applying the monotonic logarithmic function and the Gaussian model previously introduced, the optimization problem can be simplified into a nonlinear least square problem:

$$X^{MAP} = \underset{X}{\operatorname{argmin}} - \log\prod_{i=1}^{N} l(z_i|X) = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{N} \|h_i(X) - z_i\|^2_{\Sigma_i} \tag{3.74}$$

where

$$\|h_i(X) - z_i\|^2_{\Sigma_i} = (h_i(X) - z_i)^\top \Sigma_i^{-1} (h_i(X) - z_i) \tag{3.75}$$

is the Mahalanobis distance.

The nonlinear problem can be solved through standard methods, such as the Gauss-Newton or the Levenberg-Marquardt algorithms, which iteratively converge to the solution by solving the linear approximation of the nonlinear system. Given an initial state estimate X^0, the linearized measurement equation is:

$$h_i(X) = h_i(X^0 + \Delta X) \approx h_i(X^0) + H_i\Delta X \tag{3.76}$$

$$H_i = \left. \frac{\partial h_i(X)}{\partial X} \right|_{X^0} \tag{3.77}$$

where $\Delta X = X - X^0$ is the state update. Substituting this approximation in Eq. 3.74 and managing the terms, the system can be solved in function of the variable ΔX.

$$\Delta X^* = \underset{\Delta X}{\operatorname{argmin}} \sum_{i=1}^{N} \|A_i \Delta X - b_i\|^2 \tag{3.78}$$

where $A_i = \Sigma_i^{-\frac{1}{2}} H_i$ and $b_i = \Sigma_i^{-\frac{1}{2}} \big(z_i - h_i(X^0)\big)$. The current update ΔX^* is employed to propagate the estimation, whose value is used as linearization point for the next algorithm step. More information can be found in [40], [64].

3.3.3 Sensor modeling for SLAM-based strategies

The mathematical modeling of the factors used to represent the measurement constraints to solve the autonomous navigation and mapping problem is presented. It is necessary to highlight that, as opposed to the mathematical modeling employed in the UKF framework, the SLAM problem has not been modeled by using a one-to-one correspondence between sensors and factors. Inspired by [54], [65], the factors described below have been employed, where it is necessary to consider that the information included in some factors can be derived from measurements not coming from a single sensor. The state of the system at instant i is defined as a complete pose belonging to SE(3), which can be expressed mathematically as:

$$T_{x_i} = \begin{bmatrix} R_i & \boldsymbol{t}_i \\ \mathbf{0}^{1\times 3} & 1 \end{bmatrix} \tag{3.79}$$

where $R_i \in \mathrm{SO}(3)$ is the rotation matrix and $\boldsymbol{t}_i \in \mathbb{R}^3$ represents the translation vector. Defining the set of poses at time k with $\mathcal{X}_k$, such that $\mathcal{X}_k = \{T_{x_i}\}_{i=0,1,\ldots,k}$, it is possible to define the optimization problem and, in particular, Eq. 3.75 on the smooth manifold SE(3). Considering a transformation from the state x_i to the state x_j constrained with an odometry measurement $z_{i,j}$ with covariance $\Sigma_{i,j}$, Eq. 3.75 becomes:

$$\|f_{ij}(x_i, x_j) \ominus z_{i,j}\|^2_{\Sigma_{i,j}} = \| \log(T^{-1}_{z_{i,j}} T^{-1}_{x_i} T_{x_j})\|^2_{\Sigma_{i,j}} \tag{3.80}$$

The symbol $\ominus$ encodes the logarithmic map from the manifold to an element of the SE(3) Lie algebra, where $f_{ij}(\cdot)$ represents the measurement function applied to the poses T_{x_i} and T_{x_j}. For ease of explanation T_{x_i} can be represented with the vector $\left[\begin{array}{cccccc} X_{x_i} & Y_{x_i} & Z_{x_i} & \phi_{x_i} & \theta_{x_i} & \psi_{x_i} \end{array}\right] \in \mathbb{R}^6$ and the measurement function becomes

$$f_{ij}(x_i, x_j) = \left[\begin{array}{cccccc} X_{x_{i,j}} & Y_{x_{i,j}} & Z_{x_{i,j}} & \phi_{x_{i,j}} & \theta_{x_{i,j}} & \psi_{x_{i,j}} \end{array}\right]^\top \tag{3.81}$$

In contrast, for a measurement z_i that indicates a local information on the state x_i with covariance Σ_i, Eq. 3.75 is

$$\|f_i(x_i) \ominus z_i\|^2_{\Sigma_i} = \|\log(T^{-1}_{z_i} T_{x_i})\|^2_{\Sigma_i} \tag{3.82}$$

where the measurement function $f_i(\cdot)$ applied to the pose T_{x_i} can be defined as:

$$f_i(x_i) = \left[\begin{array}{cccccc} X_{x_i} & Y_{x_i} & Z_{x_i} & \phi_{x_i} & \theta_{x_i} & \psi_{x_i} \end{array}\right]^\top \tag{3.83}$$

Position and orientation binary 4D XYZ-Y factor. A relative 4D pose-to-pose constraint has been defined on x, y, and z translation and yaw rotation. The measurement information for the factor, which is denoted as XYZ-Y factor, are derived from the DVL and from the attitude estimation filter described in [66]. The function in the optimization problem can be defined as

$$f_{XYZ-Y}(x_{i-1}, x_i) = \|m_{XYZ-Y}(x_{i-1}, x_i) \ominus o_{i-1,i}\|^2_{\Sigma_{o_{i-1,i}}} \tag{3.84}$$

where $m_{XYZ-Y}(\cdot)$ and $\Sigma_{o_{i-1,i}}$ are respectively the measurement function and the covariance associated with the XYZ-Y factor and $o_{i-1,i}$ represent, on SE(3), the observation for the XYZ-Y part. In particular, the measurement function can be defined as in Eq. 3.85.

$$m_{XYZ-Y}(x_{i-1}, x_i) = \left[\begin{array}{cccc} X_{x_{i-1,i}} & Y_{x_{i-1,i}} & Z_{x_{i-1,i}} & \psi_{x_{i-1,i}} \end{array}\right]^\top \tag{3.85}$$

Orientation unary 2D RP factor. An unary 2D factor, indicated as RP factor, has been employed on roll and pitch rotations. As for the yaw data, the measurement information are provided by the attitude estimator developed in [66], where the data coming from an IMU and a FOG are fused, and the function in the optimization problem can be defined as

$$f_{RP}(x_i) = \|m_{RP}(x_i) \ominus r_i\|^2_{\Sigma_{r_i}} \tag{3.86}$$

where $m_{RP}(\cdot)$ and Σ_{r_i} are respectively the measurement function and the covariance associated with the RP factor and r_i represent, on SE(3), the observation for the RP part. In particular, the measurement function can be defined as in Eq. 3.87.

$$m_{RP}(x_i) = \begin{bmatrix} \phi_{x_i} & \theta_{x_i} \end{bmatrix}^\top \tag{3.87}$$

Position unary 1D Z factor. An unary 1D factor has been employed on z translation. The measurement is provided by the DS and the function in the optimization problem can be defined as

$$f_Z(x_i) = \|m_Z(x_i) - z_i\|^2_{\Sigma_{z_i}} \tag{3.88}$$

where $m_Z(\cdot)$ and Σ_{z_i} are respectively the measurement function and the covariance associated with the Z factor and $z_i \in \mathbb{R}$ is the depth measurement. In particular, the measurement function can be defined as in Eq. 3.89.

$$m_Z(x_i) = [Z_{x_i}] \tag{3.89}$$

Position unary 2D XY factor. An unary 2D factor has been employed as constraint on x and y translation exploiting the GPS observations, when the vehicle in on the sea surface. The function to be optimized can be defined as

$$f_{XY}(x_i) = \|m_{XY}(x_i) - \boldsymbol{g}_i\|^2_{\Sigma_{g_i}} \tag{3.90}$$

where $m_{XY}(\cdot)$ and Σ_{g_i} are respectively the measurement function and the covariance associated with the XY factor and $\boldsymbol{g}_i \in \mathbb{R}^2$ is the GPS measurement. In particular, the measurement function can be defined as in Eq. 3.91.

$$m_{XY}(x_i) = \begin{bmatrix} X_{x_i} & Y_{x_i} \end{bmatrix}^\top \tag{3.91}$$

Position and orientation binary 6D XYZ-RPY factor. A relative 6D complete pose-to-pose constraint has been defined on x, y and z translation and on roll, pitch and yaw rotation thanks to the measurement coming from the monocular camera. The measurement information for the factor, which is denoted as XYZ-RPY factor, are derived from the comparison of the feature translation induced on the image plane of two subsequent keyframes by the robot motion. The function in the optimization problem can be defined as

$$f_{XYZ-RPY}(x_i, x_j) = \|m_{XYZ-RPY}(x_i, x_j) \ominus p_{i,j}\|^2_{\Sigma_{p_{i,j}}} \tag{3.92}$$

where $m_{XYZ-RPY}(\cdot)$ and $\Sigma_{p_{i,j}}$ are respectively the measurement function and the covariance associated with the XYZ-RPY factor and $p_{i,j}$ represent, on SE(3), the observation for the XYZ-RPY part. In particular, the measurement function can be defined as in Eq. 3.93.

$$m_{XYZ-RPY}(x_i, x_j) = \begin{bmatrix} X_{x_{i,j}} & Y_{x_{i,j}} & Z_{x_{i,j}} & \phi_{x_{i,j}} & \theta_{x_{i,j}} & \psi_{x_{i,j}} \end{bmatrix}^\top \tag{3.93}$$

Camera-based landmark factor. Thanks to the features extracted from optical images and matched through multiple keyframes, it is possible to optimize map point locations $\boldsymbol{P}_j \in \mathbb{R}^3$ and keyframe poses $T_{x_i} \in$ SE(3) minimizing the reprojection error with respect to the matched keypoints $\boldsymbol{p}_{ij} \in \mathbb{R}^2$. The error term for the observation of a map point j in a keyframe i is

$$\boldsymbol{e}_{ij} = \boldsymbol{p}_{ij} - \pi_i(T_{x_i}\boldsymbol{P}_j) \tag{3.94}$$

where $\pi_i(\cdot)$ is the projection function:

$$\pi_i(T_{x_i}\boldsymbol{P}_j) = \begin{bmatrix} f_x \frac{x_{ij}}{z_{ij}} + c_x \\ f_y \frac{y_{ij}}{z_{ij}} + c_y \end{bmatrix} \tag{3.95}$$

where (f_x, f_y) and (c_x, c_y) are respectively the focal length and the principal point of the camera and $\begin{bmatrix} x_{ij} & y_{ij} & z_{ij} \end{bmatrix}^\top$ are the coordinates of the point. The cost function to be minimized can be defined as:

$$f_{LM}(x_i) = \rho\left(\|\boldsymbol{p}_{ij} - \pi_i(T_{x_i}\boldsymbol{P}_j)\|^2_{\Sigma_{lm_i}}\right) \tag{3.96}$$

where $\rho(\cdot)$ is the Huber robust cost function and Σ_{lm_i} is the covariance matrix associated to the scale at which the keypoint i was detected.

Chapter 4

Orientation initialization algorithm

A three-axis magnetometer measures the direction and strength of the total magnetic field around the device. However, it cannot distinguish between the Earth's magnetic field and additive magnetic disturbances. The direction of the terrestrial magnetic field vector depends on the geographic location; however, in the operational area of an underwater vehicle, it can be considered constant with respect to the NED frame. Magnetic disturbances conceptually fall into two different categories: external (environmental) disturbances and disturbances that rotate with the sensor. With regard to the latter, they can be further characterized as hard iron or soft iron disturbances. Permanent magnets and magnetized objects, such as electronic subsystems in the proximity of the sensor, give rise to the hard iron effect. These objects are the source of a permanent magnetic field, constant in all directions, whose effect is the addition of a constant bias on the magnetometer output of the error model. The soft iron effect is caused by ferromagnetic materials close to the sensor, such as iron and nickel, which produce a local magnetic field whose magnitude is related to the angle of incidence of Earth's magnetic field on the material itself.

The navigation filter works according to two parallel structures, where the first part, used for attitude estimation, is used as input for the second part, dedicated to position estimation. Specifically, the attitude estimation filter estimates roll, pitch, and yaw angles and their derivatives [66], [67], then sends the orientation values to the filter responsible for vehicle position estimation [20], [68]. The

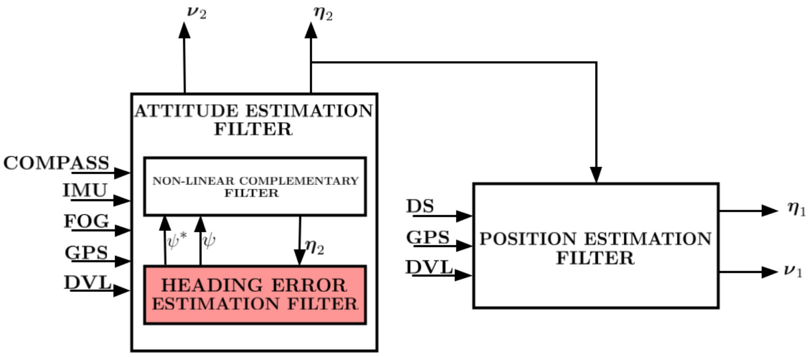

ν_2
η_2
ATTITUDE ESTIMATION FILTER
COMPASS
IMU
FOG
GPS
DVL
NON-LINEAR COMPLEMENTARY FILTER
ψ^*
ψ
η_2
HEADING ERROR ESTIMATION FILTER
DS
GPS
DVL
POSITION ESTIMATION FILTER
η_1
ν_1

Figure 4.1: Overall navigation filter framework, where the introduced heading error estimation filter is depicted in red.

heading error estimation filter works as part of the attitude estimation filter and estimates the heading error ψ^* and the heading value ψ. Only disturbances caused by objects which rigidly rotate with the sensor can be compensated, regardless of the calibration technique adopted. External metal objects are inevitably a source of magnetic disturbance that affects the yaw estimate. Their presence is not uncommon, especially in the field of underwater robotics: many AUVs are indeed used for inspection tasks of modern wrecks, mainly composed of metal parts and debris. Since these disturbances cannot be corrected, the proposed LG-EKF can provide an initial estimation of the heading error, reducing the external disturbances' effects. While at first, the only possible countermeasure was to readily detect corrupted readings and to avoid the use of the magnetometer measurements in such situations, the LG-EKF permits to continue using those measurements by introducing a corrective additive term. The proposed methods basically aims at estimating the local difference between the Magnetic North and the Geographic North directions. The Magnetic North direction is significantly deviated from the one related to the Earth magnetic field due to environmental disturbances. Supposing that, in the area of interest, the magnetic field is locally uniform, is a strong hypothesis, but, considering that the vehicle perform survey missions over the noise sources, this can be considered as acceptable. Of course, during inspection and manipulation underwater missions, where the vehicle is near to the magnetic noise sources and interacts with the surrounding environment, the assumption of uniform magnetic field cannot be exploited. Some preliminary findings of the results reported in this chapter were published in [69].

4.1 Framework development

The IMU, if the orientation is correctly initialized thanks to its magnetometers, and the FOG can provide the rotation matrix between the body and the NED frames during the whole mission. As shown in Fig. 4.1, the attitude can be computed through dedicated filters, where the IMU and the FOG measurements are appropriately combined. Considering the vehicle on the sea surface (i.e., supposing the AUV lies on a plane of the NED reference system, where the Down axis is identically equal to zero), it is possible to simplify the initialization procedure to the estimation of the yaw angle offset. This hypothesis does not reduce the generality of the proposed algorithm, thanks to the fact that the roll and pitch angle are generally estimated directly through the IMU accelerometers, which do not suffer magnetic disturbances.

The rotation matrix between the body and the NED reference systems represents the rotation to be applied to the DVL-based speed measurement, which is acquired in the body frame, to be superimposed to the GPS-based speed measurement, which is directly acquired in the NED frame. This sentence can be summarized as

$$^{N}\boldsymbol{v}_{GPS} = R_{\psi}{}^{b}\boldsymbol{v}_{DVL}, \tag{4.1}$$

where $^{N}\boldsymbol{v}_{GPS}$ and $^{b}\boldsymbol{v}_{DVL}$ are the velocities respectively measured by the GPS and the DVL and ψ is the yaw angle between the body and the NED reference frames.

When the magnetometers are not well initialized and the vehicle is on the sea surface, a rotation error between the rotated DVL speed and the GPS speed occurs, caused by an incorrect yaw estimation. Consequently, the following expression between the yaw angles can be defined:

$$R_{IMU} = R_{\psi}R_{\psi^*}, \tag{4.2}$$

where R_{IMU} is the rotation matrix where the yaw is estimated through the attitude filter based on the IMU and FOG measurements and ψ^* is the yaw offset caused by the magnetic disturbances. The effects of the magnetic disturbances have been modeled as a constant offset on the yaw angle. It can be considered an acceptable approximation due to the implementation of the attitude filter, described in [66], [67], where the roll and pitch angles are computed through the accelerometers, which are not affected by magnetic disturbances. The magnetometer measurements are employed only in the yaw angle computation, thanks to a two-step orientation estimation algorithm. As reported in [70] and [71], by employing this algorithm framework, roll and pitch have no relationship with

the magnetometer readings.
The state X is modeled to evolve on the matrix Lie group $G = \mathrm{SO}(2)^2 \times \mathbb{R}$, which can be represented as

$$X = \begin{pmatrix} R_\psi \\ R_{\psi^*} \\ \omega \end{pmatrix}_G = \begin{bmatrix} R_\psi & 0^{2\times 2} & 0^{2\times 2} \\ 0^{2\times 2} & R_{\psi^*} & 0^{2\times 2} \\ 0^{2\times 2} & 0^{2\times 2} & \begin{bmatrix} 1 & \omega \\ 0 & 1 \end{bmatrix} \end{bmatrix} \tag{4.3}$$

where ψ is the exact vehicle yaw angle, ψ^* is the yaw angle error caused by the magnetic disturbances and ω is the yaw angular rate. ω is an approximated value of the components r of the vector ${}^b\boldsymbol{\nu}_2$, where it is supposed that the vehicle is moving on a two-dimensional space, as the sea surface. The Lie algebra associated to the Lie group G is denoted as $\mathfrak{g} = \mathfrak{so}(2)^2 \times \mathbb{R}$. For a vector $\boldsymbol{x} = \begin{bmatrix} \psi & \psi^* & \omega \end{bmatrix}^\top$, the following holds

$$[\boldsymbol{x}]^\wedge_G = \begin{pmatrix} [\psi]^\wedge_{\mathrm{SO}(2)} \\ [\psi^*]^\wedge_{\mathrm{SO}(2)} \\ \omega \end{pmatrix}_{\mathfrak{g}} = \begin{bmatrix} [\psi]^\wedge_{\mathrm{SO}(2)} & & \\ & [\psi^*]^\wedge_{\mathrm{SO}(2)} & \\ & & [\omega]^\wedge_{\mathbb{R}} \end{bmatrix}. \tag{4.4}$$

The exponential map for such defined G is

$$\exp_G\left([\boldsymbol{x}]^\wedge_G\right) = \begin{pmatrix} R_\psi \\ R_{\psi^*} \\ \omega \end{pmatrix}_G = \begin{pmatrix} \exp_{\mathrm{SO}(2)}\left([\psi]^\wedge_{\mathrm{SO}(2)}\right) \\ \exp_{\mathrm{SO}(2)}\left([\psi^*]^\wedge_{\mathrm{SO}(2)}\right) \\ \omega \end{pmatrix}_G. \tag{4.5}$$

Since SO(2) and $\mathbb{R}$ are abelian and the Cartesian product of abelian groups is abelian, the adjoint operators are trivial

$$\mathrm{ad}_G(\boldsymbol{x}) = 0^{3\times 3} \tag{4.6}$$

$$\mathrm{Ad}_G\left(\exp_G\left([\boldsymbol{x}]^\wedge_G\right)\right) = \mathcal{I}^{3\times 3} \tag{4.7}$$

Given the previously introduced state representation, it is possible to define the system model. A constant angular rate model $\hat{\Omega}_k = \Omega(X_k) : G \to \mathbb{R}^3$ is employed and can be defined as

$$\Omega(X_k) = \begin{bmatrix} T\omega_k \\ 0 \\ 0 \end{bmatrix} \tag{4.8}$$

The prediction step can be performed through the following equations for the state and covariance evolution. It is possible to retrieve that

$$\mu_{k+1|k} = \mu_{k|k} \begin{bmatrix} R_{T\omega_k} & & \\ & \mathcal{I}^{2\times2} & \\ & & \mathcal{I}^{2\times2} \end{bmatrix} \tag{4.9}$$

$$P_{k+1|k} = \mathcal{F}_k P_{k|k} \mathcal{F}_k^\top + \Phi_G(\hat{\Omega}_k) Q_k \Phi_G(\hat{\Omega}_k)^\top \tag{4.10}$$

with

$$\mathcal{F}_k = \mathcal{I}^{3\times3} + \mathcal{C}_k = \begin{bmatrix} 1 & 0 & T \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

$$\Phi_G(\hat{\Omega}_k) = \mathcal{I}^{3\times3}. \tag{4.12}$$

The measurement equation can be defined by considering the on board available sensors. In particular, the GPS and the DVL-based speed measurements are compared to estimate the yaw orientation error. To constrain the yaw error estimation the measurements coming from an IMU, which suffers the magnetic disturbances, and a single-axis FOG are employed. Due to their high precision, the DVL speed measurements are considered as non-stochastic input to the dynamic system, while the uncertainty of the GPS speed measurements is taken into account in the measurement covariance matrix. We define as ${}^{b}\boldsymbol{v}_{DVL}$ and ${}^{N}\boldsymbol{v}_{GPS}$ the speed measured by the DVL and the GPS respectively, R_{IMU} the rotation matrix computed with the yaw angle estimated by the IMU under magnetic disturbances and ω_{FOG} the angular rate measured by the FOG.
The measurement map $h : G \to G'$ is given as

$$h(X) = \begin{bmatrix} 1 & 0 & (R_\psi {}^{b}\boldsymbol{v}_{DVL})_x & & & \\ 0 & 1 & (R_\psi {}^{b}\boldsymbol{v}_{DVL})_y & & & \\ 0 & 0 & 1 & & & \\ & & & R_\psi R_{\psi^*} & & \\ & & & & 1 & \omega \\ & & & & 0 & 1 \end{bmatrix} \tag{4.13}$$

Following the update equations introduced in the previous sections, the measurement matrix $\mathcal{H}_{k+1}$ can be computed as (see from Eq. 3.15 to Eq. 3.17):

$$\mathcal{H}_{k+1} = \begin{bmatrix} R_{\psi_{k+1}}[1]^{\wedge}_{\mathrm{SO}(2)}\,{}^{b}\boldsymbol{v}_{DVL} & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.14}$$

and it guarantees to easily evaluate the Kalman gain thanks to Eq. 3.25. The state and covariance update can be computed as:

$$\mu_{k+1|k+1} = \mu_{k+1|k} \begin{bmatrix} R_{\nu_{\psi}} & & \\ & R_{\nu_{\psi^*}} & \\ & & \begin{bmatrix} 1 & \nu_{\omega} \\ 0 & 1 \end{bmatrix} \end{bmatrix} \tag{4.15}$$

$$P_{k+1|k+1} = \left(\mathcal{I}^{3\times 3} - K_{k+1}\mathcal{H}_{k+1}\right) P_{k+1|k} \tag{4.16}$$

where

$$\Phi_G(\nu_{k+1}) = \mathcal{I}^{3\times 3} \tag{4.17}$$

$$\boldsymbol{\nu}_{k+1} = \begin{bmatrix} \nu_{\psi} \\ \nu_{\psi^*} \\ \nu_{\omega} \end{bmatrix} = K_{k+1} \begin{bmatrix} {}^{N}\boldsymbol{v}_{GPS} - R_{\psi_{k+1}}\,{}^{b}\boldsymbol{v}_{DVL} \\ \mathrm{atan2}(R^{\top}_{\psi^*_{k+1}} R^{\top}_{\psi_{k+1}} R_{IMU}) \\ \omega_{FOG} - \omega_{k+1} \end{bmatrix} \tag{4.18}$$

4.2 Algorithm validation

To provide evidence that the heading estimator filter can be applied in arbitrary directions, multiple straight lines in different directions, at different distances to targets causing potential disturbances, have been performed. This validation is necessary to assert that the basic idea of the filter, where it is assumed that the magnetic disturbance effect can be simplified with an offset in the heading, is an acceptable approximation. From a theoretical point of view, the implementation of the LG-EKF algorithm has been justified in light of the implementation of the attitude filter, where roll and pitch angles are computed through the accelerometers, which have no relationship with the magnetometer readings [70], [71]. Turning to the practice, a validation test was performed during an underwater mission accomplished in La Spezia (Italy) in April 2022. FeelHippo AUV, after a straight line over the sea surface to initialize the heading offset value, has performed four underwater straight lines at different depths, of various lengths and along different directions. Fig. 4.2 reports the trajectory followed

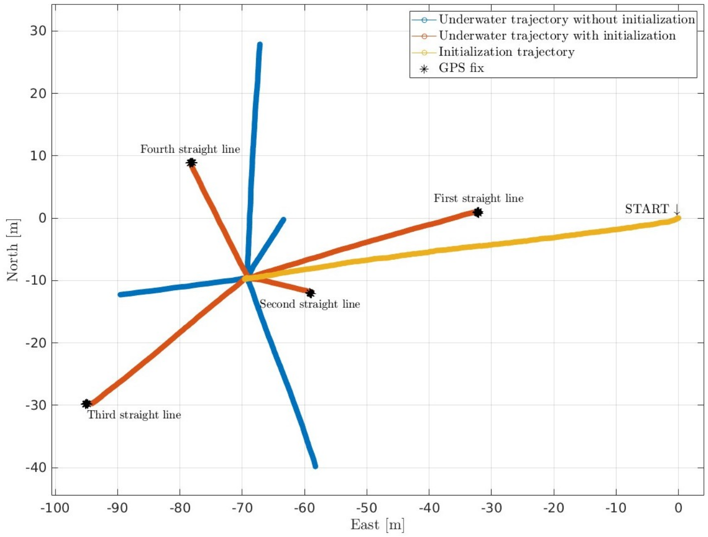


Figure 4.2: Comparison of the estimated underwater trajectories with and without the initialization procedure during the validation test performed in La Spezia (Italy). The GPS fix is employed as proof of the correctness of the estimated yaw offset.

during the initialization procedure and the underwater estimated trajectories with and without the heading correction. To guarantee the graph readability, the path followed by the vehicle over the sea surface to return to the starting point is not reported. The resurfacing error obtained thanks to a comparison between the first GPS fix and the last underwater estimated position guarantees to evaluate the goodness of the proposed strategy and its independence from parameters, as the moving direction and the depth, which influence the distance from the magnetic disturbance sources. The threshold value on the heading offset covariance to apply the heading correction to the navigation filter has been empirically set equal to 0.003 rad. When the heading offset covariance estimated through the filter is lower than the threshold value, the correction is applied to the attitude estimation filter. Fig. 4.3 reports the heading error ψ^* estimate over time and highlights the moment in which the estimated heading offset has been applied to the navigation filter. The results regarding the filter validation are summarized in Tab. 4.1.

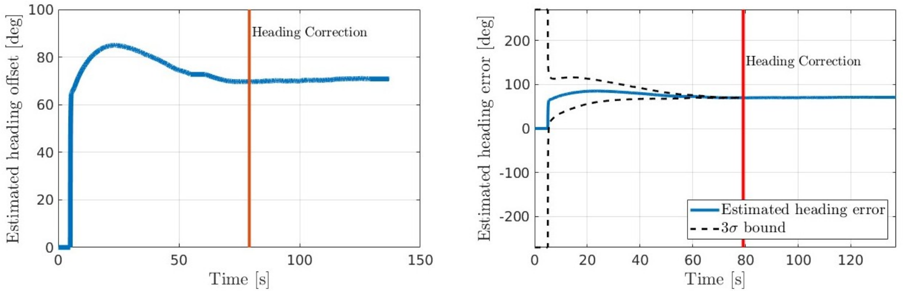


Figure 4.3: On the left, heading offset estimated through the LG-EKF during the validation test performed in La Spezia (Italy). On the right, 3σ evaluation of the heading offset estimated through the LG-EKF during the validation test performed in La Spezia (Italy). The red line indicates when the heading correction is applied to the navigation filter.

Table 4.1: LG-EKF-based initialization procedure validation experiment results.

	Resurfacing error with initialization procedure [m]	Resurfacing error without initialization procedure [m]	Filter convergence time [s]
First straight line	0.65	43.57	79.14
Second straight line	0.30	12.71	
Third straight line	0.64	37.81	
Fourth straight line	0.29	23.94	

4.3 Experimental results

The data obtained during two test campaigns have been used to evaluate the proposed LG-EKF as an initialization procedure. In particular, while the first acquisition was conducted near Cecina, Livorno (Italy), in September 2021, the second was performed at the CSSN basin in La Spezia (Italy), in April 2022. During the sea trials near Cecina, FeelHippo AUV has been employed to inspect the shipwreck of the Melania, a ship that sank about fifty years ago. Due to the presence of the shipwreck, the magnetometer data have been influenced by unknown magnetic disturbances, and consequently, the vehicle yaw has been characterized by an unknown offset. On the contrary, the sea trials conducted at the CSSN basin have been performed in a harbor, where several docked ships and submerged metal objects modify the magnetometer measurements.

During the first step of the initialization procedure to estimate the heading offset, the AUV navigated over the sea surface to acquire the sensor data for the LG-EKF. During this phase, an operator can manually guide the vehicle, or it can follow a fixed trajectory. To verify the robustness of the LG-EKF against random vehicle movements, both options have been investigated: the vehicle has been moved as ROV during the initialization procedure during the Melania monitoring mission and as AUV over a straight line during the mission accomplished in La Spezia. When the heading offset covariance estimated through the LG-EKF was lower than a fixed threshold, the AUV yaw angle was corrected. During the mission in the Cecina sea, the vehicle has been switched from ROV to AUV modality and has autonomously performed the desired underwater mission. The trajectories estimated with and without the heading offset have been compared. GPS readings were collected before FeelHippo AUV dove and after it resurfaced. They have been employed as ground truth to compute the resurfacing error and evaluate the proposed strategy. The resurfacing error value, representing an estimation of the navigation drift, has been analyzed to assess the improvements obtained thanks to the initialization procedure.

Fig. 4.4 and Fig. 4.6 report the trajectory followed during the initialization procedure and the underwater estimated trajectories with and without the heading correction. In Fig. 4.4, it can be immediately noticed that the presence of the shipwreck strongly influences the magnetometer measurements, making them unusable for attitude initialization. The resurfacing error computed by comparing the first GPS fix and the last underwater estimated position guarantees to evaluate the goodness of the proposed strategy. Analyzing the data acquired during the mission performed in La Spezia, the error caused by the magnetic disturbances is appreciable, but it is limited. On the contrary, it is necessary to notice that, during the mission in Cecina, only with the initialization procedure, the resurfacing error reaches an acceptable value, where only the drift caused by the DVL measurements integration is taken into account. The initialization procedure becomes fundamental to inspect the shipwreck autonomously, otherwise, the vehicle is unaware of its position when submerged. As for the validation experiment, the threshold value on the heading offset covariance to apply the heading correction to the navigation filter has been empirically set equal to 0.003 rad. Fig. 4.5 and Fig. 4.7 report the heading error ψ^* estimate over time for both missions. It can be noticed that when the initialization procedure is manually conducted, the requested covariance value is reached for the first time after about a minute of navigation. On the contrary, when the vehicle performs an autonomous straight line over the sea surface, the convergence is more rapid and requires about 25 seconds. In the presented results, the initialization procedure lasted more than four times of the convergence necessary time to evaluate the filter's consistency. In fact, in both cases, the LG-EKF settles on the heading error value, which is passed to the navigation filter. The main results regarding the filter convergence and consistency are reported in Tab. 4.2.

Table 4.2: LG-EKF-based initialization procedure results.

	Resurfacing error with initialization procedure [m]	Resurfacing error without initialization procedure [m]	Filter convergence time [s]
Cecina sea trial	1.76	66.03	60.37
La Spezia sea trial	0.38	5.87	22.69

4.4 Main contributions

A strategy based on LG-EKF for AUVs orientation initialization in the presence of magnetic disturbances, making magnetometer measurements unreliable, has been developed and tested. The filter has been designed to work with the available on board navigation sensors to estimate the heading angle. An initialization systematic procedure, to be performed when the vehicle is on the sea

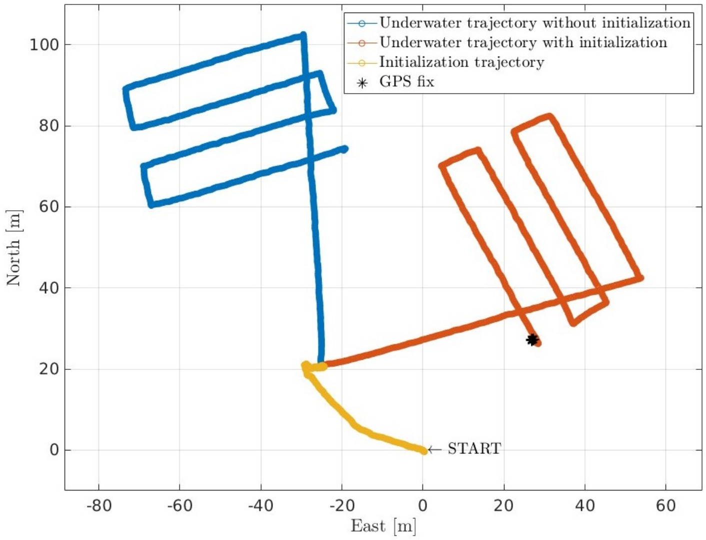


Figure 4.4: Comparison of the estimated underwater trajectories with and without the initialization procedure during the test performed in Cecina, Livorno (Italy). The GPS fix is employed as proof of the correctness of the estimated yaw offset.

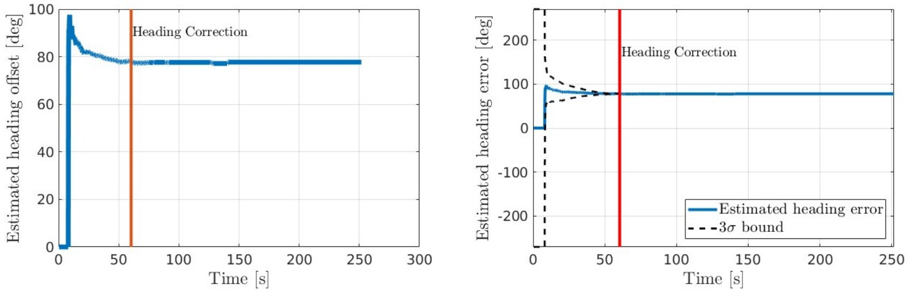


Figure 4.5: On the left, heading offset estimated through the LG-EKF during the validation test performed in Cecina, Livorno (Italy). On the right, 3σ evaluation of the heading offset estimated through the LG-EKF during the validation test performed in Cecina, Italy. The red line indicates when the heading correction is applied to the navigation filter.

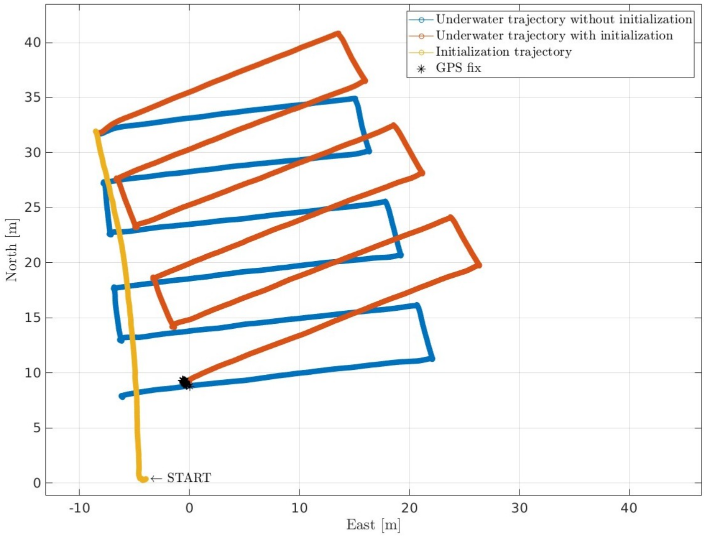


Figure 4.6: Comparison of the estimated underwater trajectories with and without the initialization procedure during the test performed in La Spezia (Italy). The GPS fix is employed as proof of the correctness of the estimated yaw offset.

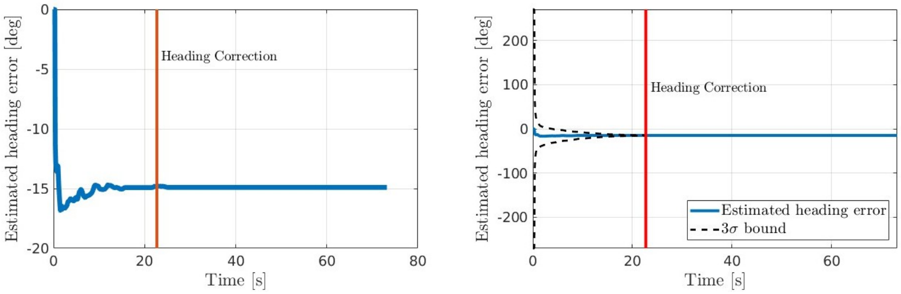


Figure 4.7: On the left, heading offset estimated through the LG-EKF during the validation test performed in La Spezia (Italy). On the right, 3σ evaluation of the heading offset estimated through the LG-EKF during the validation test performed in La Spezia (Italy). The red line indicates when the heading correction is applied to the navigation filter.

surface, has been proposed to replace the magnetometer measurements and to evaluate the offset caused by the unknown disturbances.

Chapter 5

UKF-based fusion strategies for multisensor navigation

Navigation techniques that employ multiple devices can provide a high improvement in the estimation quality, but they can also cause an increase in terms of computational load. Consequently, strategies that can represent a trade-off between these two conflicting goals have been investigated. In particular, two different frameworks have been implemented and compared: on the one hand, a centralized iterative UKF-based navigation approach and, on the other hand, a sensor fusion framework with parallel local UKFs. The main findings reported in this chapter were published in [72].

5.1 Strategies for AUV speed computation

5.1.1 DVL-based dead reckoning

Standard DR strategies usually estimate the AUV position by integrating the linear velocity measured through highly-accurate instruments, as DVLs. Assuming a discrete-time system, the mathematical equation that can be extracted from the previously introduced statement is reported below:

$$\boldsymbol{\eta}_{1,k} = \boldsymbol{\eta}_{1,k-1} + R_b^N(\boldsymbol{\eta}_{2,k-1})\boldsymbol{\nu}_{1,k-1}\Delta T \tag{5.1}$$

where $\boldsymbol{\eta}_{1,k}$ and $\boldsymbol{\eta}_{1,k-1}$ are respectively the current and previous position estimation output of the DR navigation system expressed in the NED reference system, $R_b^N(\boldsymbol{\eta}_{2,k-1})$ represents the rotation matrix between the NED and the body-fixed frames, $\boldsymbol{\nu}_{1,k-1}$ contains the body-fixed frame linear velocities and ΔT is the fixed sampling time.

5.1.2 Visual odometry

Optical sensors are commonly employed in several underwater monitoring and patrolling tasks, such as 2D mosaicing [73], automatic target recognition [74], 3D mapping, and sea bottom reconstruction [75], but, nowadays, their exploitation in navigation strategies is widely investigated to guarantee a multitasking use of the available payload and a reduction of the required on-board sensors. Mono Visual Odometry (VO) is a navigation technique that employs one camera rigidly attached to a vehicle for its motion estimation by analyzing the induced motion on the acquired images. For its nature, a VO strategy works well only if particular conditions are verified, such as a scene uniformly illuminated and with a high number of features. As input for the UKF algorithms, it has been used the linear speed computed through the VO strategy proposed in [76] and [77]. Giving as input to the VO algorithm a pair of images, it returns the transformation matrix reconstructed from the displacements of their features that can be employed, supposing known the absolute image acquisition times, to compute the body speed estimation.

5.1.3 Acoustic odometry

As for optical cameras, the FLS main application fields are 2D and 3D reconstructions [78], [79], [80], automatic target recognition [81], seabed mapping [82] and navigation aiding. Focusing the attention on navigation aiding, FLS-based DR strategies have been carefully investigated. These strategies, also called Acoustic Odometry (AO), can be defined as a navigation technique that employs a FLS rigidly attached to a vehicle for motion estimation. The algorithm's core is the registration of a pair of overlapped images obtained from different viewpoints, which is performed through a phase correlation method. Considering that a generic underwater environment can present areas with high informative content and regions where the seafloor is unaltered, the solution proposed in [15] and [83] and here briefly described is suited to face mutable underwater scenarios.

5.2 UKF-based navigation filter framework

The complete pose estimation filter works by following two parallel structures, where the first part, employed for attitude estimation, is employed as input in the second part, which is dedicated to position estimation. In particular, the attitude estimation filter is based on a compass, IMU and FOG data and it estimates the roll, pitch and yaw angles and their derivatives to send then the orientation values to the filter in charge of estimating the vehicle position [66], [67]. An UKF-based estimator with a mixed kinematic-dynamic vehicle model is employed as core of the position estimation filter. In particular, by taking into account only the longitudinal dynamics, the processing unit's computational cost is strongly reduced. An exhaustive analysis of the position estimation filter is detailed in [20], [68]. As introduced in Section 3.2, the kinematic-dynamic model formulation is based on the standard notation from the SNAME [58]. According to the assumptions exploited in [20], the dynamic model presented in Eq. 3.36 can be strongly simplified, leading to Eq. 5.2 for the dynamics along the surge axis:

$$m\dot{\nu}_{1x} = \tau_{1x}(\boldsymbol{\nu}, \boldsymbol{u}) + D_x(\boldsymbol{\nu}) \tag{5.2}$$

where $\tau_{1x}(\boldsymbol{\nu}, \boldsymbol{u})$ is the component of $\boldsymbol{\tau}(\boldsymbol{\nu}, \boldsymbol{u})$ along the surge-axis of the vehicle, m is the AUV dry mass and $D_x(\boldsymbol{\nu})$ is the damping term along the surge-axis. In particular, $D_x(\boldsymbol{\nu})$ can be represented as follows:

$$D_x(\boldsymbol{\nu}) = -C_D \nu_{1x}^2 \mathrm{sgn}(\nu_{1x}) \tag{5.3}$$

where C_D is the surge-axis drag coefficient, that depends on the parameters involved in the longitudinal drag. By following the previously introduced notation, the kinematic-dynamic model is described with the following state variables:

$$\boldsymbol{x} = \left[\begin{array}{cc} {}^N\boldsymbol{\eta}_1^\top & {}^b\boldsymbol{\nu}_1^\top \end{array} \right]^\top \tag{5.4}$$

with $\boldsymbol{x} \in \mathbb{R}^6$. Considering the discrete state evolution, a mixed kinematic-dynamic model is employed to describe the AUV behavior and it can be represented as reported in Eq. 5.5:

$$\left[\begin{array}{c} {}^N\boldsymbol{\eta}_1 \\ {}^b\boldsymbol{\nu}_1 \end{array} \right]_k = \left[\begin{array}{c} {}^N\boldsymbol{\eta}_1 \\ {}^b\boldsymbol{\nu}_1 \end{array} \right]_{k-1} + \Delta T \left[\begin{array}{c} R_b^N((\boldsymbol{\eta}_2)_{k-1})({}^b\boldsymbol{\nu}_1)_{k-1} \\ \frac{\tau_{1x}(\boldsymbol{\nu}_{k-1}, \boldsymbol{u}_{k-1})}{m} - \frac{C_D \nu_{1x}^2 \mathrm{sgn}(\nu_{1x})}{m} \\ 0 \\ 0 \end{array} \right] + \boldsymbol{w}_k \tag{5.5}$$

where k is the current iteration step, ΔT is the fixed sampling time of the filter and $\boldsymbol{w}_k$ is the additive process noise (assumed zero-mean Gaussian white noise). At the k-th instant, the measurement vector $\boldsymbol{y}_k$ can employ the following measurements coming from the available sensors:

$$\boldsymbol{y}_k = \left\{ {}^N\boldsymbol{P}_{GPS}, {}^N d_{DS}, {}^b\boldsymbol{v}_{DVL}, {}^b\boldsymbol{v}_{FLS}, {}^b\boldsymbol{v}_{CAM} \right\}, \tag{5.6}$$

where each employed variable is defined in Section 3.2.3. The position filter can use GPS fixes (available before the vehicle dives or during resurfacings) depth measurements from a DS, as well as linear speed estimations, which can be provided from multiple sensors, as the DVL, the FLS through AO strategies and the camera trough VO approaches. As explained in Section 5.1, acoustic and visual odometry strategies provide a two-dimensional speed estimation vector, which belongs to the xy-plane of the body reference frame. To guarantee uniformity, only the body-fixed frame surge and sway velocities provided by the DVL are taken into account in the algorithms. For the sake of completeness, it is necessary to highlight that the depth may be independently estimated using the DVL itself or a distinct depth sensor. In the context of this work, since the current navigation system exploits a depth sensor to estimate the depth, the proposed algorithms take into account just the AUV motion along the xy-plane. In conclusion, the measurement equation can be expressed as follows:

$$\boldsymbol{y}_k = H_k \boldsymbol{x}_k + \boldsymbol{l}_k \tag{5.7}$$

where matrix H_k is time-variant and it contains only 1 or 0 elements according to the presence of the corresponding measurement at the current iteration time and

$$\boldsymbol{l}_k = \left\{ \boldsymbol{\epsilon}_{GPS}, \epsilon_{DS}, \boldsymbol{\epsilon}_{DVL}, \boldsymbol{\epsilon}_{FLS}, \boldsymbol{\epsilon}_{CAM} \right\}, \tag{5.8}$$

where each employed variable is defined in Section 3.2.3. Furthermore, it is assumed that the biases of the position and speed measurements are negligible. Supposing that $\boldsymbol{l}_k \sim \mathcal{N}(0, R_k)$, the matrix R_k can be defined as a diagonal matrix composed of the following submatrices:

$$R_k = \left\{ R_{GPS}, R_{DS}, R_{DVL}, R_{FLS}, R_{CAM} \right\}, \tag{5.9}$$

where R_{GPS}, R_{DS}, R_{DVL}, R_{CAM} and R_{FLS} are respectively the measurement covariances related to GPS, DS, DVL, camera and FLS.

Since the navigation filter works at a fixed sampling time of $\Delta T = 0.1$ s, or, in other words, at an operative frequency of 10 Hz, due to the various sampling rate of the available sensors and due to the various working frequency of the odometry strategies, the structure of the measurement matrix H_k might change during each iteration and might be different for the parallel local filters. This issue has been solved in both the iterative and the parallel filtering approaches. Regarding the speed sources, it is finally necessary to highlight that the measurements are provided according to the sensor acquisition and elaboration rate and, for the odometry strategies, to the local environment properties. Due to these constraints, visual and acoustic odometries can respectively acquire images with a 10 Hz and 2 Hz rate, but speed measurements are computed with a lower rate. In particular, while the acoustic odometry can work with almost all the available acoustic images, the visual odometry algorithm's maximum work frequency is approximately 4 Hz. Finally, the DVL measurements have been guaranteed with a 5 Hz rate.

5.3 Filtering strategies

Handling several available measurements can be done by following multiple ways, but, necessarily, some of them need to be compliant for robotics applications due to the limitations imposed by the required computational cost that compromise the real-time action. As a matter of fact, there were better strategies to be followed than a standard UKF filter with a measurement vector containing all the available data provided by the onboard sensors. Increasing the measurement vector dimension causes the rise of the computational time requested for each filter iteration due to its effect on the filter matrices' size to be inverted and multiplied between each other [84].

The first proposed solution to overcome this issue is an iterative filtering strategy. While the iterative approach does not modify the prediction step, the correction step is performed one or more times depending on the available measurements. This approach, usually named sequential UKF, is desirable when a trade-off between the computational load and the reduction of discarded measurements is required.

The second proposed solution is a parallel filtering strategy. The position estimation filter is divided into two separate structures: the first one is a group of local filters, where each of them is dedicated to each single speed measurement source, and the second one is a master filter, which deals with the fusion of the estimates and covariances provided by the local filters to compute as output an optimized estimate and covariance. This approach guarantees, on the one hand, that all the available speed measurements are employed and, on the other hand, that the local estimates and covariances can be fused into a global estimate and covariance according to a chosen criterion imposed by the master filter. To guarantee the local filter parallelism and avoid one diverging due to the lack of measures, a reset function is applied at the end of each master filer iteration. The reset procedure guarantees that the local filters a-priori estimates and covariances are accorded to the master filter a-posteriori estimate and covariance, and it is necessary for the filter convergence. This approach will be deeply analyzed in the following, where the federated and the decentralized UKF strategies are explained. Even the global behavior of these strategies is similar, the internal functioning is markedly different. While in the federated UKF approach the state estimations are fused and weighted through the state covariances, in the decentralized UKF the state estimations and covariances are iteratively forced to converge to a common value depending on the accordance between each other.

For completeness, even the correction step in the sequential UKF is iteratively performed N times, where N is equal to the available velocity measurements, all the data coming from the sensors are provided as input directly into a single filter. This approach is usually referred to as centralized Kalman filtering. On the contrary, the federated UKF, as the decentralized UKF, is a decentralized Kalman filter, where the global filtering job is divided among a bank of subfilters, each of which is operating on a separate subset of the complete measurement suite. In both the federated and the decentralized UKFs four fundamental assumptions are taken into account for the filter framework development:

- the state vector $\boldsymbol{x}$ is the same for all local filters and the master filter;
- there is no information sharing among the local filters;
- the errors in each measurement vector are mutually uncorrelated and, thus, the global matrix R is block diagonal;
- none of the measurement vectors are fed to the master filter directly.

It is necessary to note that there are no constraints on the dimensionality of each measurement vector passed to the local filters. This missing constraint is very helpful in a filter for underwater navigation. The measurement's availability is not guaranteed at each iteration due to unpredictable external conditions, such as bubbles, an untextured environment, or acoustic noise. The main properties of the analyzed filters, which will be described in the following subsections, are summarized in Tab. 5.1.

Table 5.1: Summary of the main properties of the filters, which will be analyzed and compared. The letter C stands for centralized strategy, the letter D stands for decentralized strategy.

Navigation strategy	C	D	Main properties
Reduced UKF	✓		$\boldsymbol{y}_k = [\ \ ({}^N\boldsymbol{P}_{GPS})^\top \quad {}^N d_{DS} \quad ({}^b\boldsymbol{v}_{DVL})^\top \vee ({}^b\boldsymbol{v}_{FLS})^\top \vee ({}^b\boldsymbol{v}_{CAM})^\top\ \ ]_k^\top$ The previously used filter, where only the last acquired speed measurement in chronological order is employed.
Standard UKF	✓		$\boldsymbol{y}_k = [\ \ ({}^N\boldsymbol{P}_{GPS})^\top \quad {}^N d_{DS} \quad ({}^b\boldsymbol{v}_{DVL})^\top \quad ({}^b\boldsymbol{v}_{FLS})^\top \quad ({}^b\boldsymbol{v}_{CAM})^\top\ \ ]_k^\top$ The standard UKF filter, where all the available measurements at the iteration k are centrally passed to the filter.
Sequential (Iterated) UKF	✓		$\boldsymbol{y}_k = [\ \ ({}^N\boldsymbol{P}_{GPS})^\top \vee {}^N d_{DS} \vee ({}^b\boldsymbol{v}_{DVL})^\top \vee ({}^b\boldsymbol{v}_{FLS})^\top \vee ({}^b\boldsymbol{v}_{CAM})^\top\ \ ]_k^\top$ During the update step, the filter iterates over each available measurement, which are passed to the filter in sequential order.
Consensus-based Decentralized UKF		✓	$\boldsymbol{y}_{i,k} = [\ \ ({}^N\boldsymbol{P}_{GPS})^\top \quad {}^N d_{DS} \quad ({}^b\boldsymbol{v}_{DVL})^\top \vee ({}^b\boldsymbol{v}_{FLS})^\top \vee ({}^b\boldsymbol{v}_{CAM})^\top\ \ ]_k^\top$ A decentralized strategy where the local estimations and covariances iteratively converge to a common value. Each local filter works as a standard UKF with only one speed measurement and the master filter fuses the local results.
Federated UKF		✓	$\boldsymbol{y}_{i,k} = [\ \ ({}^N\boldsymbol{P}_{GPS})^\top \quad {}^N d_{DS} \quad ({}^b\boldsymbol{v}_{DVL})^\top \vee ({}^b\boldsymbol{v}_{FLS})^\top \vee ({}^b\boldsymbol{v}_{CAM})^\top\ \ ]_k^\top$ A decentralized strategy, where the local filters produce their own estimates and covariances. The master filter fuses the local estimations thanks to a covariance-based weighted mean process. A reset procedure is employed to avoid local filters divergence.

5.3.1 Reduced UKF

In the reduced UKF (Fig. 5.1) the computational burden is limited by reducing the measurements that have been employed during each correction step of the filter. At the k-th instant, the measurement vector $\boldsymbol{y}_k \in \mathbb{R}^5$ is:

$$\boldsymbol{y}_k = \begin{bmatrix} ({}^N\boldsymbol{P}_{GPS})^\top & {}^N d_{DS} & ({}^b\boldsymbol{v}_{DVL})^\top \vee ({}^b\boldsymbol{v}_{FLS})^\top \vee ({}^b\boldsymbol{v}_{CAM})^\top \end{bmatrix}_k^\top . \tag{5.10}$$

Following this framework, several speed measurements can be discarded due to their redundancy at each filter iteration. As shown in Eq. 5.10 only one speed measurement is employed in the vector $\boldsymbol{y}_k$ and, if multiple ones are available, only one of them is chosen by following the chronological acquisition order. The matrix R_k depends on the actual dimension and composition of the measurement vector. This property is necessarily determined by the measurement data availability during the in progress iteration. Considering the presence of all the measurements that can be provided by the on-board sensors, the matrix R_k can be defined as:

$$\begin{aligned} R_k = \operatorname{diag}\{ \; R_{GPS}, \quad R_{DS}, \quad R_{DVL} \vee R_{FLS} \vee R_{CAM} \; \} = \\ = \begin{bmatrix} R_{GPS} & 0_{2\times1} & 0_{2\times2} \\ 0_{1\times2} & R_{DS} & 0_{1\times2} \\ 0_{2\times2} & 0_{2\times1} & R_{DVL} \vee R_{FLS} \vee R_{CAM} \end{bmatrix} \end{aligned} , \tag{5.11}$$

where the employed symbols have been defined previously. As a matter of fact, measurement redundancy is not handled and every new speed measurement, regardless of the sensor that bought it, is overwritten on the previous one. This approach can cause the loss of helpful information and, consequently, a growth of the uncertainty on the AUV state knowledge, which results in an increase of the covariance matrix.

5.3.2 Standard UKF

The standard UKF centrally fuses all the valid measurements during each filter step. By considering the measurement equation Eq. 5.7 and supposing that all the measurements (provided by GPS, DS, DVL, FLS and camera) are available, the measurement vector $\boldsymbol{y}_k$ can be expressed as a vector belonging to the $\mathbb{R}^9$ space:

$$\boldsymbol{y}_k = \begin{bmatrix} ({}^N\boldsymbol{P}_{GPS})^\top & {}^N d_{DS} & ({}^b\boldsymbol{v}_{DVL})^\top & ({}^b\boldsymbol{v}_{FLS})^\top & ({}^b\boldsymbol{v}_{CAM})^\top \end{bmatrix}_k^\top . \tag{5.12}$$

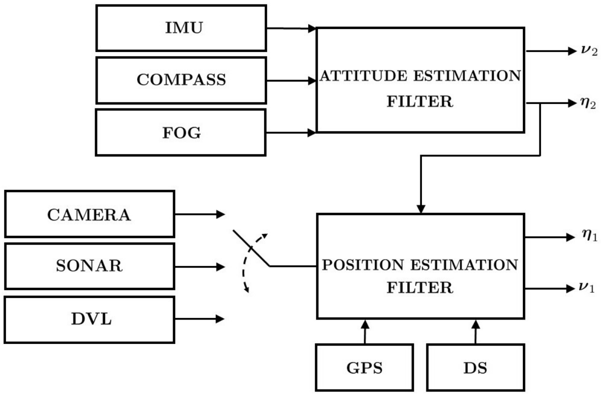


Figure 5.1: Framework of the existing navigation filter employed by the AUVs developed by the UNIFI DIEF.

All the measurements are processed at the same time, which can influence the required computational time due to the inversion of matrices with higher sizes. Considering the presence of all the measurements that can be provided by the on-board sensors, the matrix R_k can be defined as:

$$R_k = \operatorname{diag}\left\{ \; R_{GPS}, \quad R_{DS}, \quad R_{DVL}, \quad R_{FLS}, \quad R_{CAM} \; \right\}. \tag{5.13}$$

5.3.3 Sequential UKF

The sequential (or iterated) UKF (Fig. 5.2 and Fig. 5.3) approach aims to minimize the dimension of the innovation covariance matrix. This can be an important advantage for applications where the computational load is limited by the available hardware, such as in mobile underwater robotics, since such a matrix is inverted during every correction step. Suppose that instead of employing the entire measurement vector $\boldsymbol{y}_k \in \mathbb{R}^9$ (see Eq. 5.12), N_m measurement vectors $\boldsymbol{y}_k^{(i)} \in \{\mathbb{R}^1, \mathbb{R}^2\}$ are passed to the filter, where N_m is the number of the available measurements, $i = 1, 2, ..., N_m$ and k is the current iteration step. The measurement vectors that are compared are the following:

$$\boldsymbol{y}_k = \left[\; ({}^N\boldsymbol{P}_{GPS})^\top \quad {}^N d_{DS} \quad ({}^b\boldsymbol{v}_{DVL})^\top \quad ({}^b\boldsymbol{v}_{FLS})^\top \quad ({}^b\boldsymbol{v}_{CAM})^\top \; \right]_k^\top \in \mathbb{R}^9 \tag{5.14}$$

$$\boldsymbol{y}_k^{(i)} = \left[\; ({}^N\boldsymbol{P}_{GPS})^\top \vee {}^N d_{DS} \vee ({}^b\boldsymbol{v}_{DVL})^\top \vee ({}^b\boldsymbol{v}_{FLS})^\top \vee ({}^b\boldsymbol{v}_{CAM})^\top \; \right]_k^\top . \tag{5.15}$$

The employed symbols have been defined previously. Instead of processing all the measurements at time k as a greater dimension single vector, a Kalman filter that can elaborate smaller dimension multiple vectors has been implemented. Considering the presence of all the measurements that the on-board sensors can provide, the matrix $R_k^{(i)} \in \{\mathbb{R}^{1\times1}, \mathbb{R}^{2\times2}\}$ can be defined as:

$$R_k^{(i)} = R_{GPS} \vee R_{DS} \vee R_{DVL} \vee R_{FLS} \vee R_{CAM}. \tag{5.16}$$

Turning to the filter framework, it is necessary to underline that the sequential UKF prediction step is performed as in the standard UKF. The main changes can be observed in the correction stage, where the onboard sensors' measurements are employed. As for the measurement vector $\boldsymbol{y}_k^{(i)}$, in the following of this section, the here reported notation will be employed: $\hat{\boldsymbol{x}}_{k|k}^{(i)}$ is the optimal state estimate after the i-th measurement has been processed at time k and $P_{k|k}^{(i)}$ is the correspondent covariance matrix. From these definitions, it is possible to retrieve that the following equalities can be employed for correction step initialization:

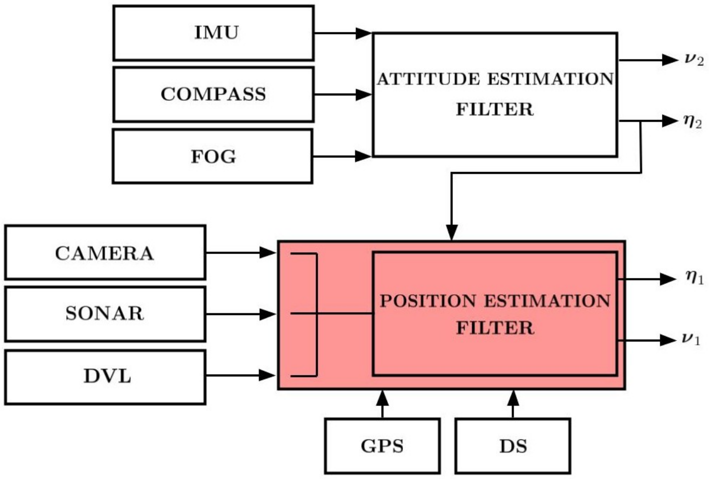


Figure 5.2: Framework of the navigation filter, where the modification introduced by the proposed sequential UKF, depicted in red, can be noticed in the position estimation filter.

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k-1}^{(0)} &= \hat{\boldsymbol{x}}_{k|k-1} \\ P_{k|k-1}^{(0)} &= P_{k|k-1} \end{aligned} \tag{5.17}$$

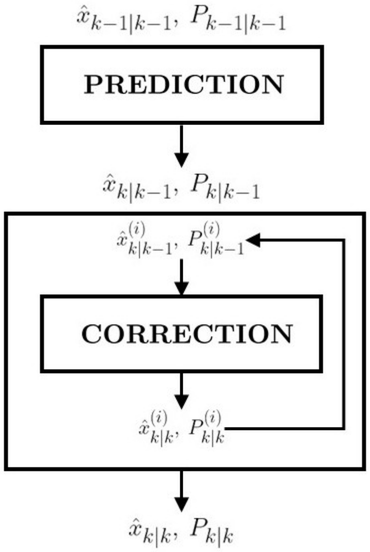

$\hat{x}_{k-1|k-1}$, $P_{k-1|k-1}$
PREDICTION
$\hat{x}_{k|k-1}$, $P_{k|k-1}$
$\hat{x}^{(i)}_{k|k-1}$, $P^{(i)}_{k|k-1}$
CORRECTION
$\hat{x}^{(i)}_{k|k}$, $P^{(i)}_{k|k}$
$\hat{x}_{k|k}$, $P_{k|k}$

Figure 5.3: Flowchart of the proposed sequential UKF.

where $\hat{\boldsymbol{x}}^{(0)}_{k|k-1}$ is the estimate after zero measurements have been processed and $P^{(0)}_{k|k-1}$ is its covariance matrix. The estimate $\hat{\boldsymbol{x}}^{(i)}_{k|k}$ and the covariance $P^{(i)}_{k|k}$ are computed by using the traditional UKF correction equations, which are repeated N_m times. After each iteration, the updated estimate and covariance are employed as initialization for the subsequent one. After all N_m measurement vectors are processed, the corrected estimate and covariance can be retrieved as:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}^{(N_m)}_{k|k} \\ P_{k|k} &= P^{(N_m)}_{k|k} \end{aligned} \quad . \tag{5.18}$$

The sequential UKF filter is summarized in Algorithm 3.

5.3.4 Consensus-based Decentralized UKF

Distributed Kalman filtering algorithms are generally employed for sensor networks, where each node can share the information flow only with its neighbors on the network. Distributed Kalman filtering strategies for fully connected sensor networks, or consensus-based decentralized Kalman filtering strategies, involve state estimation using a set of local Kalman filters that communicate with all

Algorithm 3 Sequential (or Iterated) UKF.

```
Function SEQUENTIAL_UKF():
    /* prediction step                                              */
    Step Prediction():
        Input  : $\hat{\boldsymbol{x}}_{k-1|k-1}, P_{k-1|k-1}, f_{k-1}(\cdot)$
        Output: $\hat{\boldsymbol{x}}_{k|k-1}, P_{k|k-1}$
        $(\hat{\boldsymbol{x}}_{k|k-1}, \overline{P}_{k|k-1}) = UT(\hat{\boldsymbol{x}}_{k-1|k-1}, P_{k-1|k-1}, f_{k-1}(\cdot))$;
        $P_{k|k-1} = \overline{P}_{k|k-1} + Q_{k-1}$;
    end
    /* correction step                                              */
    Step Correction():
        Input  : $\hat{\boldsymbol{x}}_{k|k-1}, P_{k|k-1}, h(\cdot)_k$
        Output: $\hat{\boldsymbol{x}}_{k|k}, P_{k|k}$
        $\hat{\boldsymbol{x}}^{(0)}_{k|k-1} = \hat{\boldsymbol{x}}_{k|k-1}$;
        $P^{(0)}_{k|k-1} = P_{k|k-1}$;
        for $i = 1, ..., N_m$ do
            $\left(\boldsymbol{y}_{k|k-1}, \bar{S}_k, P^{xy}_k\right) = UT\left(\hat{\boldsymbol{x}}^{(i)}_{k|k-1}, P^{(i)}_{k|k-1}, h_k(\cdot)\right)$;
            $S_k = \bar{S}_k + R_k$;
            $L_k = P^{xy}_k S^{-1}_k$;
            $\boldsymbol{e}^{(i)}_k = \boldsymbol{y}^{(i)}_k - \boldsymbol{y}_{k|k-1}$;
            $\hat{\boldsymbol{x}}^{(i)}_{k|k} = \hat{\boldsymbol{x}}^{(i)}_{k|k-1} + L_k \boldsymbol{e}^{(i)}_k$;
            $P^{(i)}_{k|k} = P^{(i)}_{k|k-1} - L_k S_k L^{\top}_k$;
            $\hat{\boldsymbol{x}}^{(i+1)}_{k|k-1} = \hat{\boldsymbol{x}}^{(i)}_{k|k}$;
            $P^{(i+1)}_{k|k-1} = P^{(i)}_{k|k}$;
        end
        $\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}^{(N_m)}_{k|k}$;
        $P_{k|k} = P^{(N_m)}_{k|k}$;
    end
end
```

other nodes. The distributed and decentralized Kalman filters generally use a static or dynamic consensus algorithm, which permits the fusion of estimate

and covariance data obtained by each node. The distributed filtering theory, which has been largely employed for multi-agent system applications, has been here adapted to work with a sensor network composed of the sensors devoted to speed measurement.
Within the consensus-based decentralized Kalman filter approach (Fig. 5.4 and Fig. 5.5), the information form of a single central filter for a sensor network observing a process (*i.e.*, the AUV localization process), boils down to a proper number of local filters that collectively calculate the same state estimate. The sensor network dedicated to the speed measurement can be represented as a simple fully-connected triangular graph $G(\mathcal{N}, \mathcal{E})$, where

$$\mathcal{N} = \{\mathrm{DVL}, \mathrm{CAM}, \mathrm{FLS}\} \tag{5.19}$$

is the node set composed by the DVL, the camera and the FLS and

$$\mathcal{E} = \{\{\mathrm{DVL}, \mathrm{CAM}\}, \{\mathrm{DVL}, \mathrm{FLS}\}, \{\mathrm{CAM}, \mathrm{FLS}\}\} \tag{5.20}$$

is the edge set. In the following, the generic edge that connects the i-th and the j-th sensor, chosen from the set composed of the DVL, the FLS and the camera, will be referred to as $\{i, j\}$, where $\{i, j\} \in \mathcal{E}$, $i \in \mathcal{N}$ and $j \in \mathcal{N}$. The employed consensus algorithm is reported in the information form and is based on the following discrete-time iterative equation:

$$\hat{\boldsymbol{\phi}}^{(t)}_{i,k|k} = \hat{\boldsymbol{\phi}}^{(t-1)}_{i,k|k} - \epsilon \sum_{j \neq i} a_{ij} \left(\hat{\boldsymbol{\phi}}^{(t-1)}_{i,k|k} - \hat{\boldsymbol{\phi}}^{(t-1)}_{j,k|k} \right) \tag{5.21}$$

$$\Omega^{(t)}_{i,k|k} = \Omega^{(t-1)}_{i,k|k} - \epsilon \sum_{j \neq i} a_{ij} \left(\Omega^{(t-1)}_{i,k|k} - \Omega^{(t-1)}_{j,k|k} \right) \tag{5.22}$$

where ϵ is the discretization step, which has to necessary be sufficiently reduced to guarantee the algorithm convergence, a_{ij} is the consensus coefficient between each sensors couple $\{i, j\}$, $\hat{\boldsymbol{\phi}}^{(t)}_{i,k|k} = P^{-1}_{i,k|k} \hat{\boldsymbol{x}}_{i,k|k}$ and $\Omega_{i,k|k} = P^{-1}_{i,k|k}$ are respectively the information vector and the information matrix after the correction step of the i-th filter and after the t-th iteration of the consensus algorithm. Eq. 5.21 and Eq. 5.22 can be manipulated to obtain an equivalent structure, where the autonomous and the consensus parts of the algorithm is highlighted:

$$\hat{\boldsymbol{\phi}}^{(t)}_{i,k|k} = \left(1 - \sum_{j \neq i} \epsilon a_{ij} \right) \hat{\boldsymbol{\phi}}^{(t-1)}_{i,k|k} + \sum_{j \neq i} \epsilon a_{ij} \hat{\boldsymbol{\phi}}^{(t-1)}_{j,k|k} \tag{5.23}$$

$$\Omega_{i,k|k}^{(t)} = \left(1 - \sum_{j \neq i} \epsilon a_{ij}\right) \Omega_{i,k|k}^{(t-1)} + \sum_{j \neq i} \epsilon a_{ij} \Omega_{j,k|k}^{(t-1)}. \tag{5.24}$$

The discrete-time collective dynamics of the network driven by Eq. 5.23 and Eq. 5.24 can be written as:

$$\hat{\boldsymbol{\phi}}_{k|k}^{(t)} = \mathcal{P} \hat{\boldsymbol{\phi}}_{k|k}^{(t-1)} \tag{5.25}$$

$$\Omega_{k|k}^{(t)} = \mathcal{P} \Omega_{k|k}^{(t-1)} \tag{5.26}$$

with $\mathcal{P} = I - \epsilon\mathcal{L}$ Perron matrix of a graph G with parameter ϵ, which component can be retrieved with

$$\mathcal{P}_{ij} = \begin{cases} 1 - \epsilon \sum_{j \neq i} a_{ij}, & \text{if } i = j \\ \epsilon a_{ij}, & \text{if } i \neq j \end{cases}. \tag{5.27}$$

The matrix $\mathcal{L}$ is the weighted Laplacian matrix. The consensus coefficient a_{ij} is a weight coefficient that is updated after each consensus iteration, which has been designed to consider not only the covariance matrix, but also the accordance between each local state estimation for the global one computation. Practically, each a_{ij} coefficient is defined through Eq. 5.28.

$$a_{ij} = \frac{1}{1 + n_{ij}} \tag{5.28}$$

with

$$n_{ij} = \|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j\|. \tag{5.29}$$

It is immediate to notice that more the estimates $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{x}}_j$ are similar, more the coefficient a_{ij} is near to one, which is the upper limit for the weight factor. The value of the coefficient a_{ij} influences the estimate convergence to a common value, by weighting more the estimates with a reduced covariance and a high consensus coefficient.

In the consensus-based decentralized UKF approach, as in the federated UKF, the i-th local filter state vector is full-order and, at step k, is assumed to have the availability of its respective corrected estimate $\hat{\boldsymbol{x}}_{i,k|k}$ and its associated error covariance $P_{i,k|k}$, which are employed in the master filter. The input measurement vector provided to the i-th filter at time k is $\boldsymbol{y}_{i,k}$ and the measurement equation of the local filter is, as in the federated UKF:

$$\boldsymbol{y}_{i,k} = H_{i,k}\boldsymbol{x}_k + \boldsymbol{l}_{i,k} \tag{5.30}$$

where $\boldsymbol{l}_{i,k}$ is a zero mean random variables with covariance $R_{i,k}$. Two stop criterion has been chosen for the consensus-based decentralized UKF. The algorithm could work until:

- it reaches a fixed maximum number of iterations N_{iter};
- the mean distance between the state vector estimates provided by each node of the graph is lower than a fixed threshold γ.

In the first case, while the global covariance matrix is retrieved by computing the inverse matrix of the sum of all the local information matrices, the global state estimate is retrieved by computing a covariance-based weighted mean. Eq. 5.31 and Eq. 5.32 summarizes the previous sentence.

$$P_{m,k|k} = \left(\sum_{i=1}^{N} P_{i,k|k}^{-1}\right)^{-1} \tag{5.31}$$

$$\hat{\boldsymbol{x}}_{m,k|k} = P_{k|k}\left(\sum_{i=1}^{N} P_{i,k|k}^{-1}\hat{\boldsymbol{x}}_{i,k|k}\right). \tag{5.32}$$

In the second stop condition, which is the most frequent due to the convergence speed of the consensus algorithm, the global state estimate and its covariance matrix can be set equal to one of the correspondent value taken from one local filter. It is not important the chosen local filter employed during this step, because all the local estimates and covariances converged to the same values under a threshold, which value has been a-priori set with the aim of guaranteeing at least a fifth order convergence. The reset procedure is not necessary for the consensus-based UKF. The consensus algorithm guarantees the convergence of all the local estimates and covariances, which are employed as initialization data for the following step. The consensus-based decentralized UKF filter is summarized in Algorithm 4.

5.3.5 Federated UKF

The federated UKF scheme employs the principle of information sharing among the local filters handled by the master filter. As introduced before, the federated

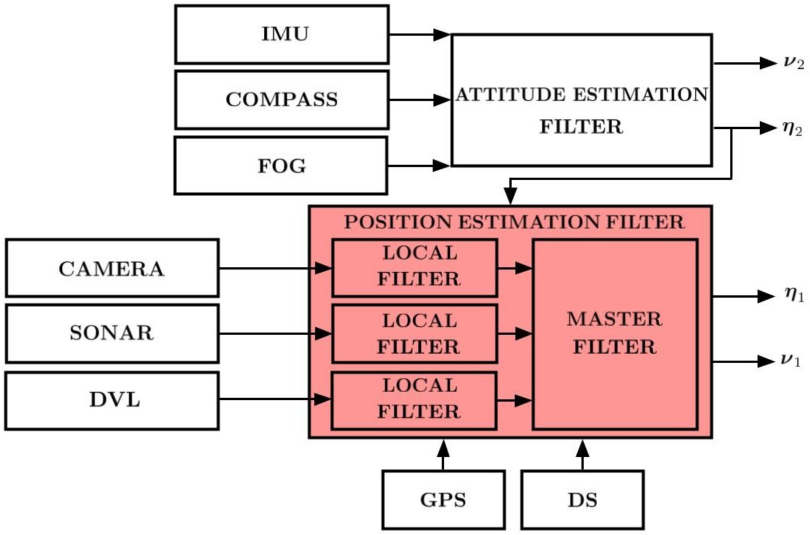


Figure 5.4: Framework of the navigation filter, where the modification introduced by the proposed consensus-based decentralized UKF, depicted in red, can be noticed in the position estimation filter.

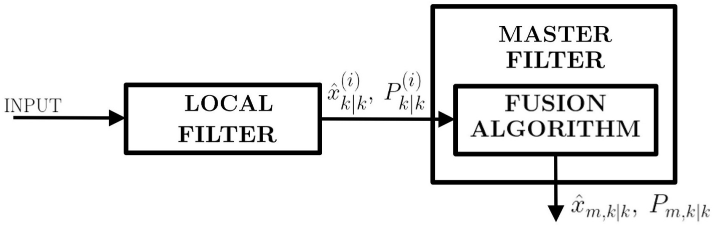

INPUT
LOCAL FILTER
$\hat{x}^{(i)}_{k|k}$, $P^{(i)}_{k|k}$
MASTER FILTER
FUSION ALGORITHM
$\hat{x}_{m,k|k}$, $P_{m,k|k}$

Figure 5.5: Flowchart of the proposed consensus-based decentralized UKF.

Algorithm 4 Consensus-based Decentralized UKF.

```
/* Consensus step of the decentralized UKF                                  */
Step Consensus():
    Input  : $\{\hat{\boldsymbol{x}}^{(i)}_{k|k}\}_{i=1,2,...,N}, \{P^{(i)}_{k|k}\}_{i=1,2,...,N}$
    Output: $\hat{\boldsymbol{x}}_{m,k|k}, P_{m,k|k}$
    ParametersUpdate()
     while $t \leq N_{iter}$ do
        for $i = 1, ..., N$ do
            $\Omega^{(t)}_{i,k|k} = \Omega^{(t-1)}_{i,k|k} - \epsilon \sum_{j \neq i} a_{ij} \left( \Omega^{(t-1)}_{i,k|k} - \Omega^{(t-1)}_{j,k|k} \right)$ ;
            $P_{i,k|k} = \Omega^{-1}_{i,k|k}$;
            $\hat{\boldsymbol{\phi}}^{(t)}_{i,k|k} = \hat{\boldsymbol{\phi}}^{(t-1)}_{i,k|k} - \epsilon \sum_{j \neq i} a_{ij} \left( \hat{\boldsymbol{\phi}}^{(t-1)}_{i,k|k} - \hat{\boldsymbol{\phi}}^{(t-1)}_{j,k|k} \right)$ ;
            $\hat{\boldsymbol{x}}_{i,k|k} = P_{i,k|k} \hat{\boldsymbol{\phi}}_{i,k|k}$;
        end
        ParametersUpdate()
         Stop()
    end
end
```

UKF approach is based on a double-step data processing architecture, in which a master filter subsequently combines the outputs of the sensor-related local filters (Fig. 5.6 and Fig. 5.7). In the federated UKF approach, the i-th local filter is assumed to implement the full-order state vector and, at step k, is assumed to have the availability of its respective prior estimate $\hat{\boldsymbol{x}}_{i,k|k-1}$ and its associated error covariance $P_{i,k|k-1}$. The input measurement vector provided to the i-th filter at time k is $\boldsymbol{y}_{i,k}$ and, as explained in Eq. 5.33, the measurement equation of the local filter is:

$$\boldsymbol{y}_{i,k} = H_{i,k} \boldsymbol{x}_k + \boldsymbol{l}_{i,k} \tag{5.33}$$

where $\boldsymbol{l}_{i,k}$ is a zero-mean random variables with covariance $R_{i,k}$. By assuming that the local filters do not have access to each other measurements, they form their respective corrected estimates and covariances according to the traditional UKF equations. It is important to notice that the local estimates are optimal for the locally provided measurements, but not with respect to all the available measurements, due to the local filter independence hypothesis.

Algorithm 5 Stop criterion and parameters update.

```
/* Stop criterion                                                */
Function Stop():
    Input: $\{\hat{\boldsymbol{x}}_{k|k}^{(i)}\}_{i=1,2,...,N}, t$
    if $t = N_{iter}$ then
        $P_{m,k|k} = \left(\sum_{i=1}^{N} P_{i,k|k}^{-1}\right)^{-1}$;
        $\hat{\boldsymbol{x}}_{m,k|k} = P_{k|k}\left(\sum_{i=1}^{N} P_{i,k|k}^{-1}\hat{\boldsymbol{x}}_{i,k|k}\right)$;
        return
    end
    if $\sum_{\{i,j\}} \|\hat{\boldsymbol{x}}_{i,k|k}^{(t)} - \hat{\boldsymbol{x}}_{j,k|k}^{(t)}\| \leq \gamma$ then
        $P_{m,k|k} = P_{i,k|k}$;
        $\hat{\boldsymbol{x}}_{m,k|k} = \hat{\boldsymbol{x}}_{i,k|k}$;
        return
    end
end
/* Consensus parameters update                                   */
Function ParametersUpdate():
    Input: $\{\hat{\boldsymbol{x}}_{k|k}^{(i)}\}_{i=1,2,...,N}$
    for $i = 2, ..., N$ do
        for $j = 1, ..., i-1$ do
            $n_{ij} = \|\hat{\boldsymbol{x}}_{i,k|k} - \hat{\boldsymbol{x}}_{j,k|k}\|$;
            $a_{ij} = \frac{1}{1+n_{ij}}$;
        end
    end
end
```

Turning the attention to the master filter, which is an UKF dedicated to the computation of an optimal global estimate of the state vector $\boldsymbol{x}$, the following notation is employed: $\hat{\boldsymbol{x}}_{m,k|k-1}$ is the optimal estimate of $\boldsymbol{x}$ conditioned on all the measurement vector provided to the local filters up to but not including $\boldsymbol{y}_{i,k}$ and $P_{m,k|k-1}$ is the associated covariance matrix. The optimal global estimate and the corresponding error covariance, which are reported in the information form of the Kalman filter, can be retrieved as:

$$P^{-1}_{m,k|k} = P^{-1}_{m,k|k-1} + \sum_{i=1}^{N} \left(H^{\top}_{i,k} R_{i,k} H_{i,k} \right) \tag{5.34}$$

$$\hat{\boldsymbol{x}}_{m,k|k} = P_{m,k|k} \left(P^{-1}_{m,k|k-1} \hat{\boldsymbol{x}}_{m,k|k-1} + \sum_{i=1}^{N} \left(H^{\top}_{i,k} R_{i,k} \boldsymbol{y}_{i,k} \right) \right) \tag{5.35}$$

where the incorrelation of the measurement vectors hypothesis is employed. However, Eq. 5.34 and Eq. 5.35 have to be modified in light of one of the master filter properties, which implies that the master filter does not have directly access to the measurement vectors $\boldsymbol{y}_{i,k}$. As consequence, Eq. 5.34 and Eq. 5.35 have to be rewritten in terms of the local filters estimates and covariances. It results that:

$$P^{-1}_{m,k|k} = P^{-1}_{m,k|k-1} + \sum_{i=1}^{N} \left(P^{-1}_{i,k|k} - P^{-1}_{i,k|k-1} \right) \tag{5.36}$$

$$\hat{\boldsymbol{x}}_{m,k|k} = P_{m,k|k} \left(P^{-1}_{m,k|k-1} \hat{\boldsymbol{x}}_{m,k|k-1} + \sum_{i=1}^{N} \left(P^{-1}_{i,k|k} \hat{\boldsymbol{x}}_{i,k|k} - P^{-1}_{i,k|k-1} \hat{\boldsymbol{x}}_{i,k|k-1} \right) \right). \tag{5.37}$$

The federated UKF filter is summarized in Algorithm 6, where the correction step is performed by following the information Kalman filtering approach, which is based on the definition of the information matrix $\Omega_{m,k|k} = P^{-1}_{m,k|k}$ vector $\hat{\boldsymbol{\phi}}_{m,k|k} = P^{-1}_{m,k|k} \hat{\boldsymbol{x}}_{m,k|k}$.

A reset procedure is applied to guarantee the information transfer from the master filter to the local filters. At the end of each iteration, the local filter estimates and covariances are updated with the global estimate and covariance. This last step is performed by using the equations:

$$\hat{\boldsymbol{x}}_{i,k|k} = \hat{\boldsymbol{x}}_{m,k|k} \tag{5.38}$$

$$P_{i,k|k} = \gamma_i P_{m,k|k} \tag{5.39}$$

for $i = 1, 2, ..., N$ and with γ_i such that $\sum_{i=1}^{N} \gamma_i^{-1} = 1$. As no information to weight the local filters are available, the γ_i factors have been set equal for all the local filters [85]. As the local filters can make their own local projections and repeat the cycle at step $k+1$, the master filter projects the global estimate and covariance to obtain the a-priori information for the following iteration. Thus,

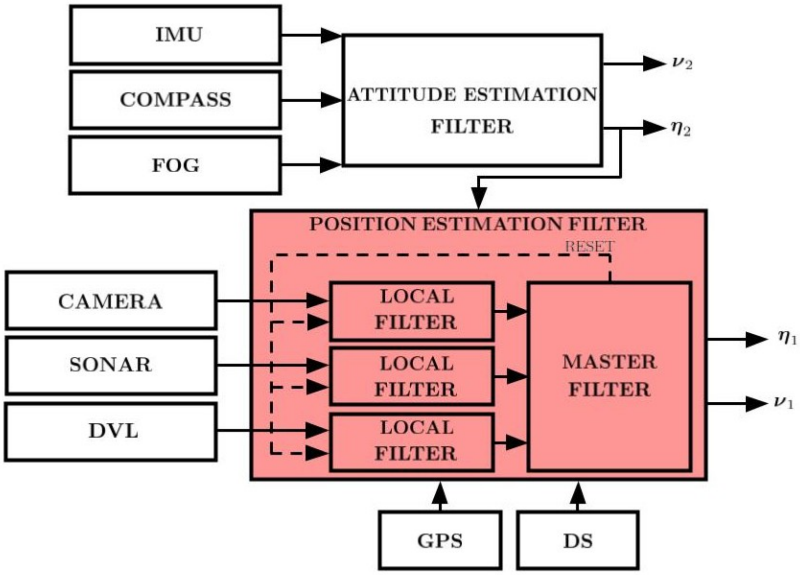


Figure 5.6: Framework of the navigation filter, where the modification introduced by the proposed federated UKF, depicted in red, can be noticed in the position estimation filter.

the multiple filters architecture permits complete autonomy of local filters and yields optimal local estimates with respect to the available local measurements. Furthermore, the master filter achieves optimality in the global estimate. This solution permits maintaining the independence of the local filters between each other and simultaneously achieving the optimal global solution.

5.4 Validation in simulated environment

Some experiments in a simulated environment have been performed to highlight the properties of the proposed filters. The main goal of these simulations is to evaluate the consistency of the analyzed strategies and how they propagate the estimated covariances. The position filter was fed with GPS position measurements during the simulations, when the vehicle was higher than a fixed depth, depth measurements, and multiple speed measurements. To increase adherence to the real dataset, the speed measurements have been generated at a different fixed rate for each sensor, by employing frequency values for DVL-based DR,

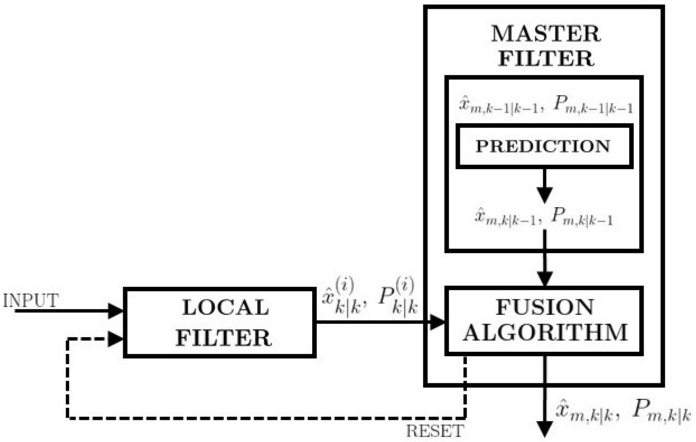

MASTER FILTER
$\hat{x}_{m,k-1|k-1}$, $P_{m,k-1|k-1}$
PREDICTION
$\hat{x}_{m,k|k-1}$, $P_{m,k|k-1}$
INPUT
LOCAL FILTER
$\hat{x}^{(i)}_{k|k}$, $P^{(i)}_{k|k}$
FUSION ALGORITHM
$\hat{x}_{m,k|k}$, $P_{m,k|k}$
RESET

Figure 5.7: Flowchart of the proposed federated UKF.

VO and AO adherent to reality. Following the frequency values obtained as an approximate availability rate for the speed data in the real-world experiments, the DVL measurements have been generated with a 5 Hz rate, the camera-based and FLS-based speed estimation, respectively, at 2 Hz and 1 Hz. To focus the attention only on the filter capabilities, instead of applying the VO and AO algorithms to estimate the camera and FLS-based velocities, all the speed measurements have been generated as random variables defined as:

$$\begin{aligned} {}^{b}\boldsymbol{v}_{DVL} &= {}^{b}\boldsymbol{\nu}_1 + \mathcal{N}(\mathbf{0}, \boldsymbol{\delta}) \\ {}^{b}\boldsymbol{v}_{CAM} &= {}^{b}\boldsymbol{\nu}_1 + \mathcal{N}(\mathbf{0}, \boldsymbol{\delta}) \\ {}^{b}\boldsymbol{v}_{FLS} &= {}^{b}\boldsymbol{\nu}_1 + \mathcal{N}(\mathbf{0}, \boldsymbol{\delta}) \end{aligned} \tag{5.40}$$

where ${}^{b}\boldsymbol{\nu}_1$ is the speed true value and $\mathcal{N}(\mathbf{0}, \boldsymbol{\delta})$ is the uncertainty defined as a normal distribution with zero-mean and covariance $\boldsymbol{\delta} = \mathrm{diag}\left\{\ 0.1,\quad 0.05\ \right\}$. The surge speed has been set equal to 0.5 m/s, and the covariance values have been chosen as a consequence. The proposed strategies have been tested on a vehicle whose dynamic behavior has been simulated by using the equations defined in [58] and that has traveled a rectangular path at a fixed depth of 2 m. For each filter, a Monte Carlo simulation with 100 iterations has been performed. The noise affecting the three sensors has been modeled to have the same characterisation from a statistical point of view. This hypothesis, which has been employed only for the simulation experiments, has been assumed to

Algorithm 6 Federated UKF.

```
Function FEDERATED_UKF():
    /* prediction step                                                    */
    Step Prediction():
        Input  : $\hat{\boldsymbol{x}}_{m,k-1|k-1}, P_{m,k-1|k-1}, f_{k-1}(\cdot)$
        Output: $\hat{\boldsymbol{x}}_{m,k|k-1}, P_{m,k|k-1}$
        $(\hat{\boldsymbol{x}}_{m,k|k-1}, \overline{P}_{m,k|k-1}) = UT(\hat{\boldsymbol{x}}_{m,k-1|k-1}, P_{m,k-1|k-1}, f_{k-1}(\cdot))$;
        $P_{m,k|k-1} = \overline{P}_{m,k|k-1} + Q_{k-1}$;
    end
    /* fusion step                                                        */
    Step Fusion():
        Input  : $\hat{\boldsymbol{x}}_{m,k|k-1}, P_{m,k|k-1}, \{\hat{\boldsymbol{x}}_{i,k|k-1}\}_{i=1,2,\dots,N}, \{P_{i,k|k-1}\}_{i=1,2,\dots,N}$,
                 $\{\hat{\boldsymbol{x}}_{i,k|k}\}_{i=1,2,\dots,N}, \{P_{i,k|k}\}_{i=1,2,\dots,N}$
        Output: $\hat{\boldsymbol{x}}_{m,k|k}, P_{m,k|k}$
        $\Omega_{m,k|k} = P^{-1}_{m,k|k-1} + \sum_{i=1}^{N} \left( P^{-1}_{i,k|k} - P^{-1}_{i,k|k-1} \right)$;
        $P_{m,k|k} = \Omega^{-1}_{m,k|k}$;
        $\hat{\boldsymbol{\phi}}_{m,k|k} = P^{-1}_{m,k|k-1} \hat{\boldsymbol{x}}_{m,k|k-1} + \sum_{i=1}^{N} \left( P^{-1}_{i,k|k} \hat{\boldsymbol{x}}_{i,k|k} - P^{-1}_{i,k|k-1} \hat{\boldsymbol{x}}_{i,k|k-1} \right)$;
        $\hat{\boldsymbol{x}}_{m,k|k} = P_{m,k|k} \hat{\boldsymbol{\phi}}_{m,k|k}$;
    end
    /* reset procedure                                                    */
    Step Reset():
        Input  : $\hat{\boldsymbol{x}}_{m,k|k}, P_{m,k|k}$
        Output: $\{\hat{\boldsymbol{x}}_{i,k|k}\}_{i=1,2,\dots,N}, \{P_{i,k|k}\}_{i=1,2,\dots,N}$
        $\hat{\boldsymbol{x}}_{i,k|k} = \hat{\boldsymbol{x}}_{m,k|k}$;
        $P_{i,k|k} = \gamma_i P_{m,k|k}$;
    end
end
```

focus the analysis only on the filter properties. Following this idea, the influence of the particular strategy used to evaluate the speed measurements to feed the filter has been removed and the analyzed filters have been compared for their intrinsic properties. The position errors and the estimated 3σ bounds along the East and North directions are reported from Fig. 5.8 to Fig. 5.12. All the 3σ bounds continuously diverge when the vehicle is under the sea surface and no position measurements are available. It correctly represents the behavior of the AUV, which has no access to any new absolute position information until it resurfaces. Furthermore, it can be noticed that all the filters, except the consensus-based distributed UKF, (e.g., reduced UKF, standard UKF, sequential UKF and federated UKF) have a similar 3σ bound propagation. The consensus-based distributed UKF 3σ bounds have the same behavior as the other filters, but its divergence is reduced. This can be related to the particular fusion strategy employed in the filter, which is dependent on the accordance between each local estimation. Comparing the 3σ bounds estimated through the filter and the standard deviation computed from the data, it is possible to assert that the all the filters have a conservative behaviour. Indeed, except for consensus-based distributed UKF, where the filter and data-based covariance is similar, the estimated 3σ bounds diverge more than the computed covariance during the underwater phase of the simulations. Furthermore, the estimated resurfacing position has been compared with the theoretical first GPS fix and its 3σ bound (Fig. 5.13). It is necessary to notice that the resurfacing positions estimated in all the Monte Carlo simulations fall inside the 3σ bound, guaranteeing acceptable estimations. Furthermore, it is possible to compare the 3σ bound estimation obtained from the filters and the 3σ bound estimation obtained from the simulated data. The latter has been evaluated by computing the best normal distribution approximating the estimated resurfacing positions with respect to the theoretical ones (Fig. 5.14). Comparing the filter and the data-based estimated covariances, which have been reported for the whole mission and, in particular, for the resurfacing time, it is possible to notice that all the filters are consistent with their estimations, but the best results from this point of view can be obtained with the consensus-based decentralized UKF.

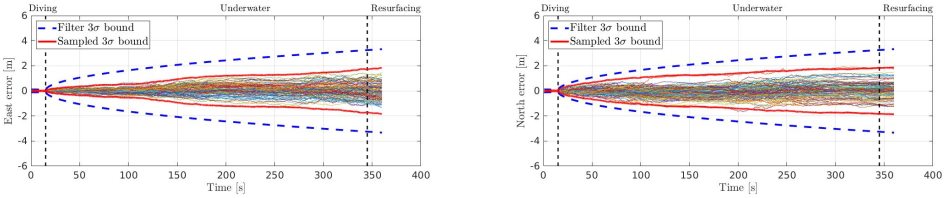


Figure 5.8: East and North position estimation errors versus their 3σ bounds obtained from 100 simulation analysis with the reduced UKF. While the sampled standard deviation is calculated from the data, the filter σ values are computed as the square-root of the corresponding diagonal element of the estimated covariance matrix.

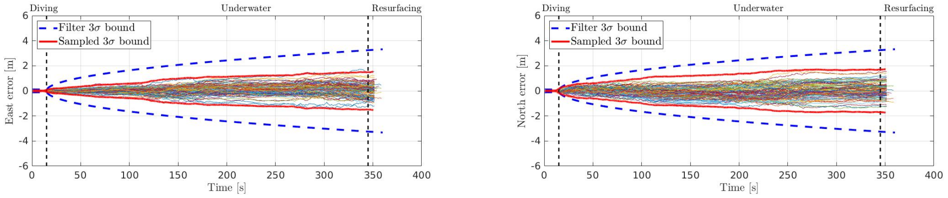


Figure 5.9: East and North position estimation errors versus their 3σ bounds obtained from 100 simulation analysis with the standard UKF. While the sampled standard deviation is calculated from the data, the filter σ values are computed as the square-root of the corresponding diagonal element of the estimated covariance matrix.

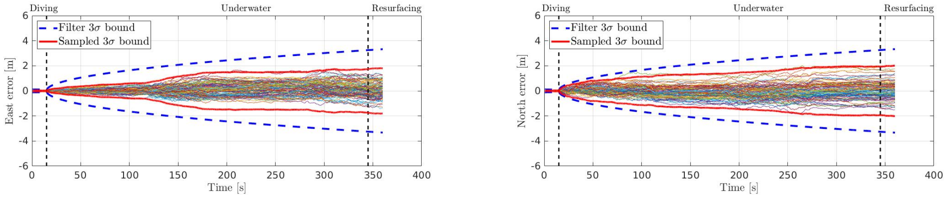


Figure 5.10: East and North position estimation errors versus their 3σ bounds obtained from 100 simulation analysis with the sequential UKF. While the sampled standard deviation is calculated from the data, the filter σ values are computed as the square-root of the corresponding diagonal element of the estimated covariance matrix.

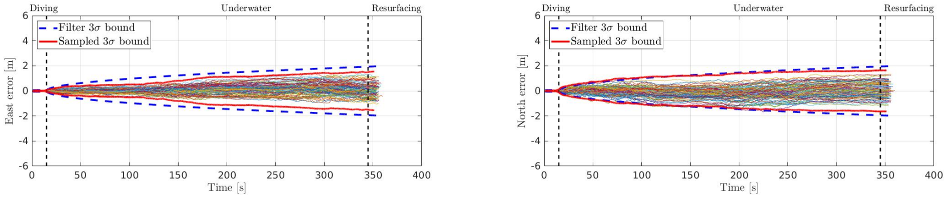


Figure 5.11: East and North position estimation errors versus their 3σ bounds obtained from 100 simulation analysis with the consensus-based decentralized UKF. While the sampled standard deviation is calculated from the data, the filter σ values are computed as the square-root of the corresponding diagonal element of the estimated covariance matrix.

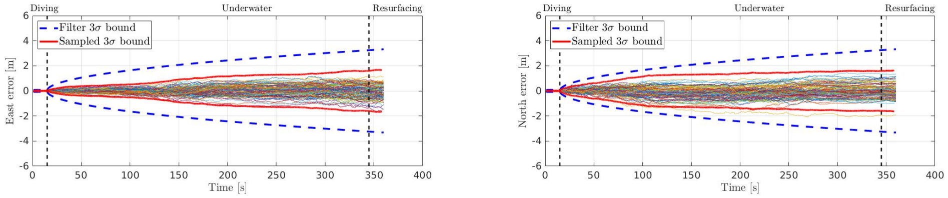
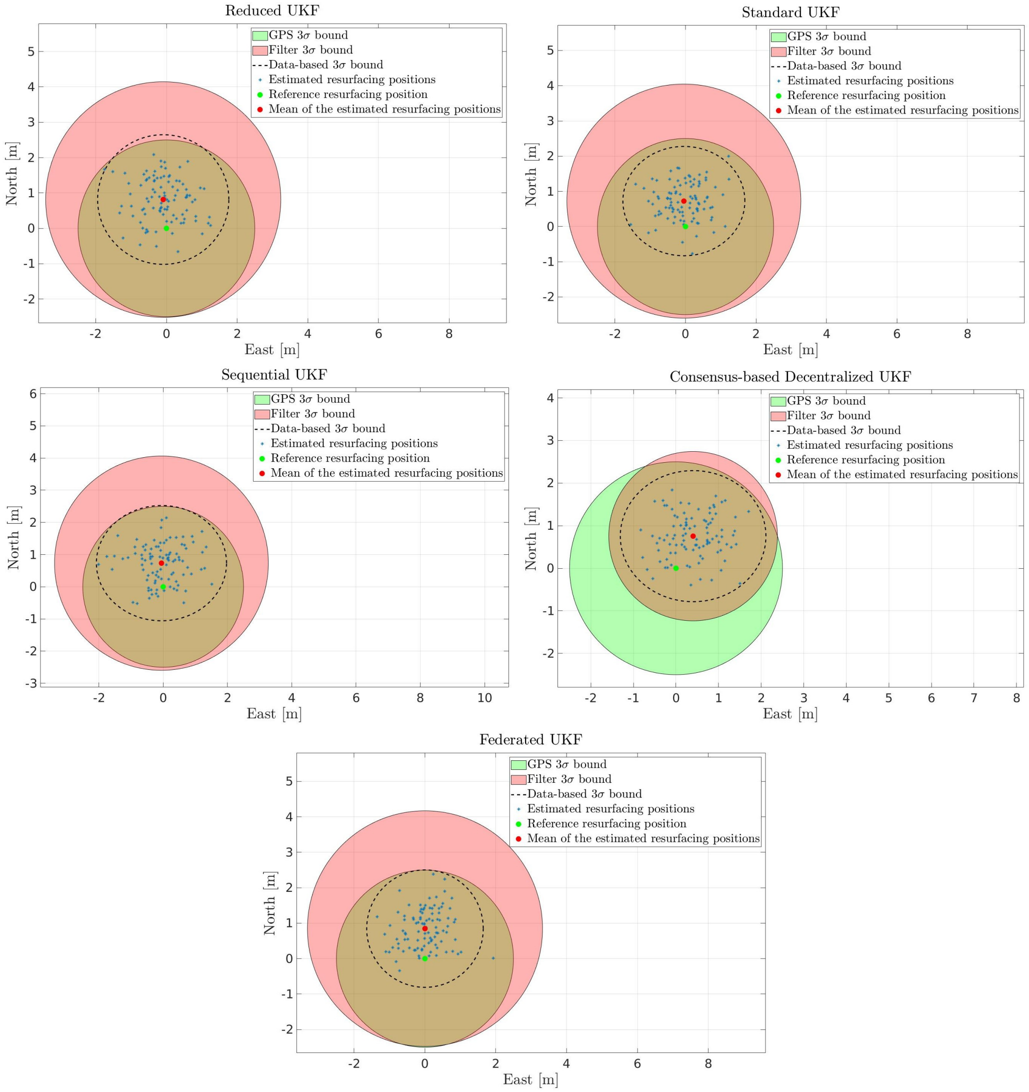


Figure 5.12: East and North position estimation errors versus their 3σ bounds obtained from 100 simulation analysis with the federated UKF. While the sampled standard deviation is calculated from the data, the filter σ values are computed as the square-root of the corresponding diagonal element of the estimated covariance matrix.

Indeed, this filter provides data-based and filter-based 3σ bounds similar estimations. Besides that, it is necessary to note that in all cases, the filter-based 3σ bounds estimations are larger than the data-based ones and, consequently, all the filters have a conservative behavior.
Turning to the speed errors and the related 3σ bounds, it can be noticed from Fig. 5.15 and Fig. 5.16 that the estimation errors remain for all the time within the bounds. It is necessary to highlight that only the output of one iteration of the Monte Carlo simulation has been reported for the speed estimations.

5.5 Experimental results

The presented navigation strategies have been tested and validated offline by employing experimental data recorded in Vulcano Island, Messina (Italy), in June 2019, during two autonomous underwater missions performed in the framework of the European project EUMR. The underwater mission was performed at a constant advance speed of 0.5 m/s with a fixed depth of 2.5 m and an altitude from the sea bottom between 2 and 4 m. In both missions, the payload sensors were switched on, and the vehicle, during its autonomous navigation along a pre-programmed path, acquired both acoustic and optical images. GPS readings were collected before FeelHippo AUV dove and after it resurfaced and they have been employed as ground truth to compute the resurfacing error and to compare the proposed strategies. As reference path, the UKF-based estimation, presented in [20], [68], is employed. Of course, considering that some measurements are discarded, this reference path cannot be employed as ground truth, but it can represent a reference whose navigation performance has to



Reduced UKF

Standard UKF

Sequential UKF

Consensus-based Decentralized UKF

Federated UKF

Figure 5.13: The estimated resurfacing positions versus the theoretical GPS fix position obtained from 100 simulation analysis for each filter.

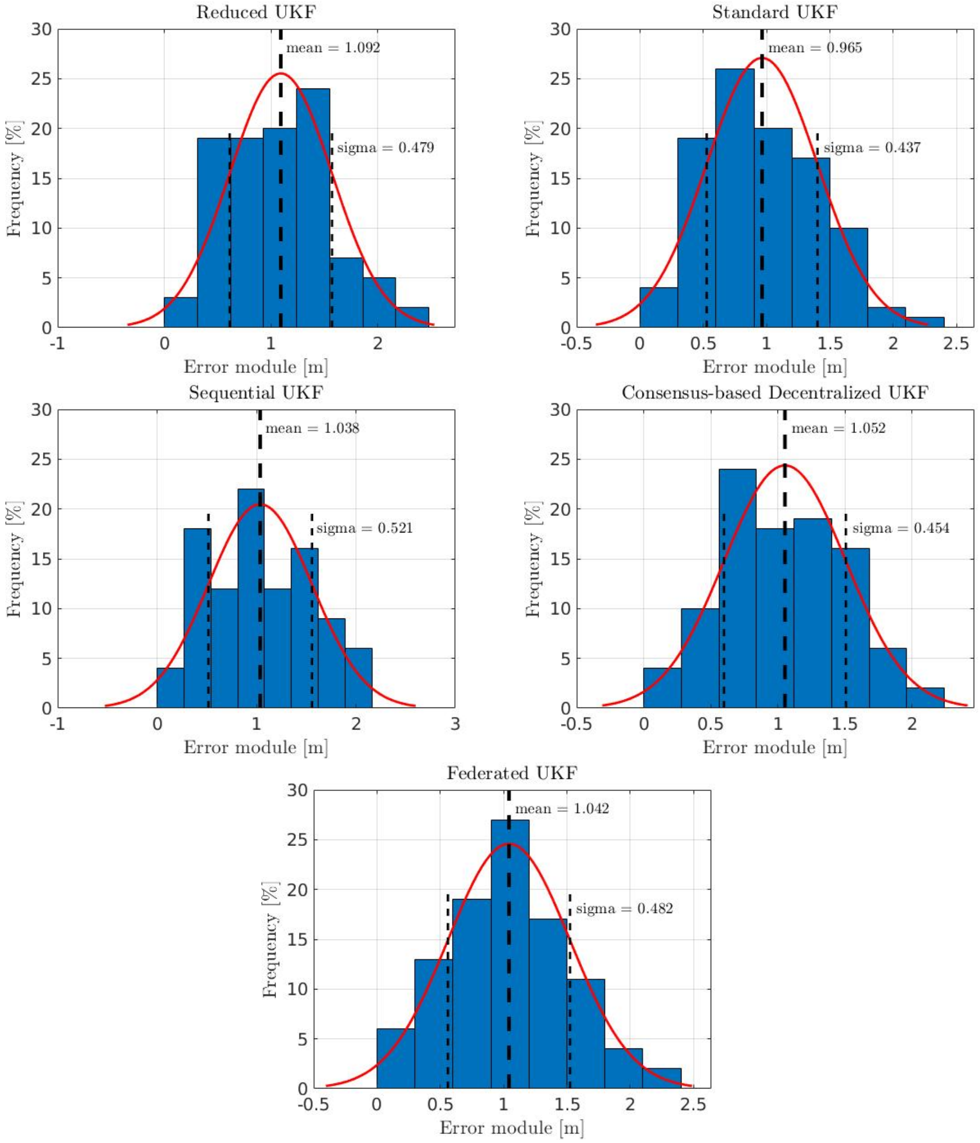


Figure 5.14: Histograms containing the estimated resurfacing position errors obtained from 100 simulation analysis for each filter.

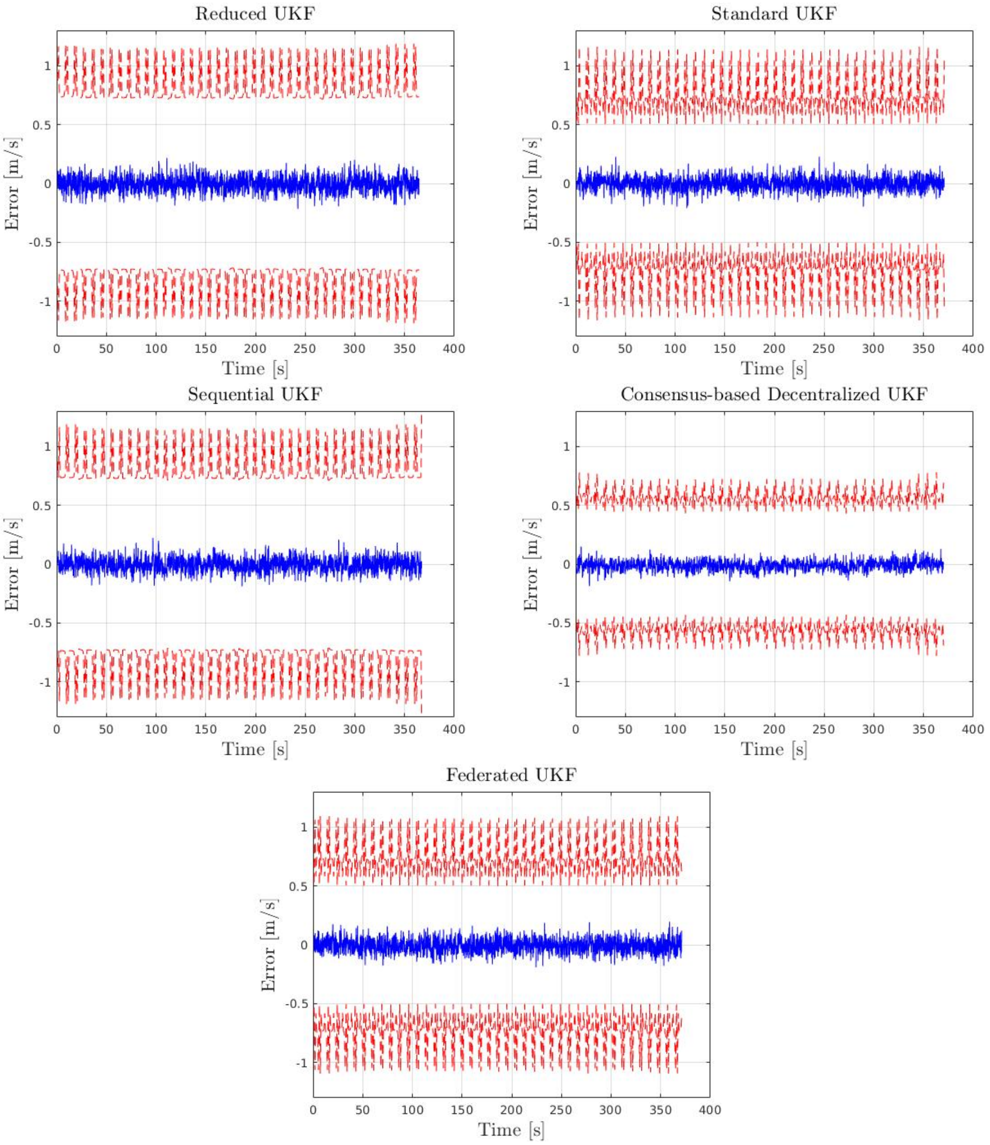


Figure 5.15: Surge speed estimation errors versus their 3σ bounds obtained from a simulation analysis. The σ values are computed as the square-root of the corresponding diagonal element of the estimated covariance matrix.

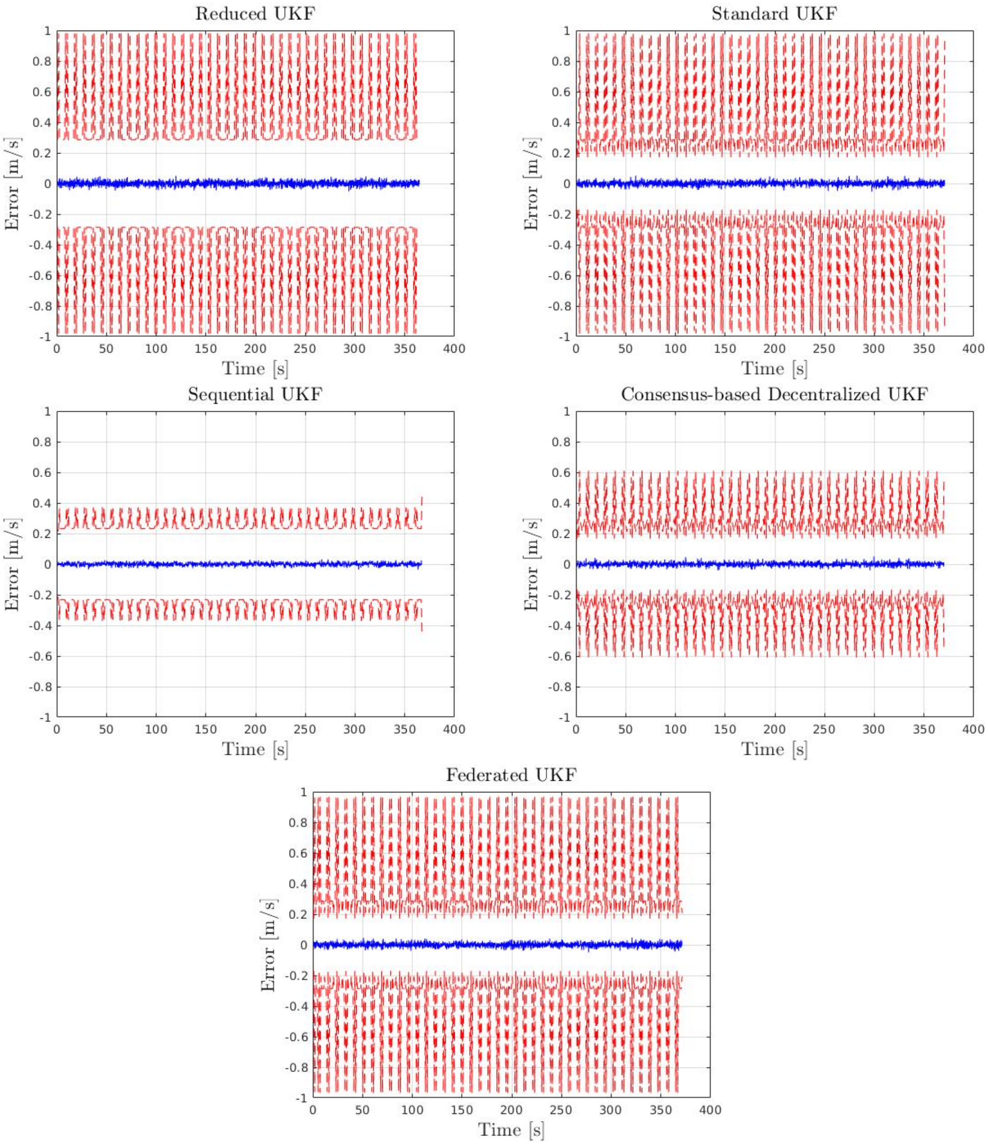


Figure 5.16: Sway speed estimation errors versus their 3σ bounds obtained from a simulation analysis. The σ values are computed as the square-root of the corresponding diagonal element of the estimated covariance matrix.

be overcome by the proposed novel approaches. The resurfacing error value, representing an estimation of the navigation drift, has been analyzed to assess the possible improvements of the presented strategies. Along the path, only the waypoints employed to plan the autonomous underwater mission can be used as a reference to evaluate the goodness of the navigation strategies.
Furthermore, an UKF which centrally employs all the available measurements has been tested to provide a reference path estimation. Considering that a centralized filter that uses at the same time all the available measurements provides the optimal estimate, the obtained trajectory can be considered as ground truth. The first autonomous mission lasted around 260 s covering approximately 100 m. From Fig. 5.17 and Tab. 5.2, it is easily noticeable that all the proposed strategies increase the navigation estimation quality, analyzing the results from the GPS resurfacing error and the planned path error point of view. The path employed for the second autonomous mission is approximately 150 m long and it has been covered in about 390 s. As for the other mission, the proposed strategies led to an improvement of the navigation performance, which can be retrieved from Fig. 5.18 and Tab. 5.3.

To evaluate the agreement between estimation errors and estimated uncertainty, the 3σ bounds of the estimated trajectories are presented. Both the analyzed missions present the same structure. In the beginning, the vehicle is on the surface and GPS fixes bound the position terms of the state covariance. Subsequently, due to the absence of position measurements, the position terms of the state covariance increase until the vehicle resurfaces. When above surface, GPS fixes are available again and the covariance gets smaller due to GPS corrections. This is summarized from Fig. 5.19 to Fig. 5.23, where, concerning the first autonomous mission, the 3σ bounds for the filter and the GPS are presented. In all the analyzed cases, the position provided by the filter (with its confidence bounds) appears to guarantee a reasonable prediction of the vehicle's true position when it resurfaces. The employed GPS has an expected accuracy on the order of meters and the 2D error can be represented as a 2D Gaussian distribution whose components are independently distributed.

Finally, it is essential to evaluate the robustness of the proposed strategies against reducing the available measurements. In the previously presented results, reported in Fig. 5.17 and Fig. 5.18, all the available speed measurements have been passed to the navigation filters. A large amount of inertial, acoustic and optical data necessarily determines the good performance of the employed navigation strategy, reducing the importance of the particular technique used to merge the measurements. To highlight the importance of a fusion-based filter as a compromise between reduced computational load and navigation robustness,

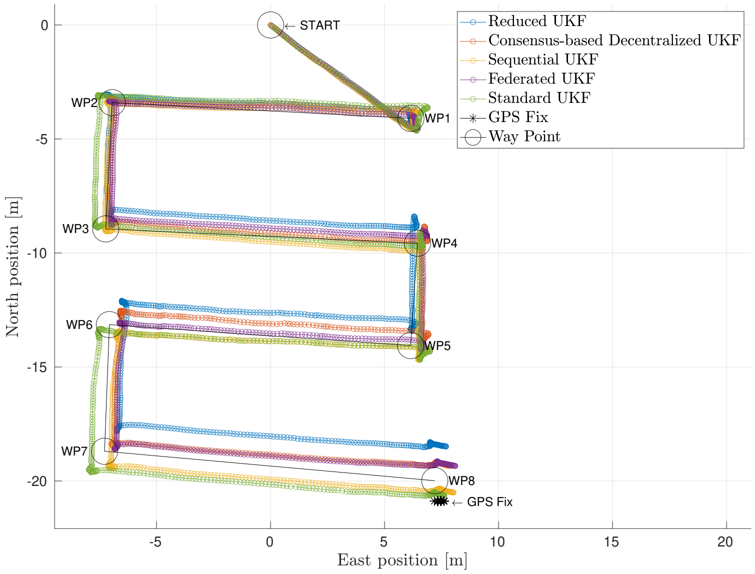


Figure 5.17: Comparison of the navigation results during mission #1. While the first underwater point is labeled with START, the GPS position acquired by the AUV after its resurfacing is identified with GPS Fix. Each waypoint of the planned autonomous mission is referred with WP.

Table 5.2: Navigation performance for the mission #1: resurfacing error.

Navigation strategy	Error [m]
Reduced UKF	2.44
Standard UKF	0.46
Sequential UKF	0.92
Consensus-based Distributed UKF	1.70
Federated UKF	1.80

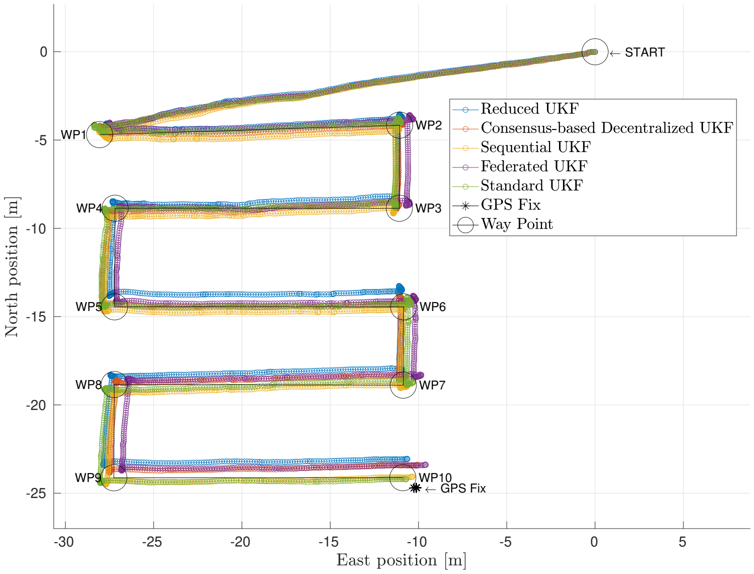


Figure 5.18: Comparison of the navigation results during mission #2. While the first underwater point is labeled with START, the GPS position acquired by the AUV after its resurfacing is identified with GPS Fix. Each waypoint of the planned autonomous mission is referred with WP.

Table 5.3: Navigation performance for the mission #2: resurfacing error.

Navigation strategy	Error [m]
Reduced UKF	1.68
Standard UKF	0.62
Sequential UKF	0.65
Consensus-based Distributed UKF	1.32
Federated UKF	1.48

the proposed UKF-based algorithms have been applied to two different subsets of the data acquired during the first mission; one reduced by 50%, the other one reduced by 75%. The selection of the measurements passed to the filters has been performed by applying a statistical selection function to each provided speed measurement. The choice has been based on the Bernoulli distribution test, whose probability mass function is reported in Eq. 5.41.

$$P(b|q) = \begin{cases} q & \text{if } b = 1 \\ 1 - q & \text{if } b = 0 \end{cases} \tag{5.41}$$

where q represents the probability to get $b = 1$ or, equivalently, a true and $1 - q$ represents the probability to get $b = 0$ or, equivalently, a false in a double-choice experiment. In the following results, the coefficient q of the Bernoulli distribution has been set to three different values, namely $q = 1$, which provides the reference path for the algorithm robustness analysis and whose results have been reported in Fig. 5.17, and $q = 0.5$ and $q = 0.25$, which impose the employment of a half and a quarter of the available measurements (Fig. 5.24 and Fig. 5.25). To give a quantitative evaluation of the robustness of the proposed algorithms, the navigation performance of the strategies applied to the

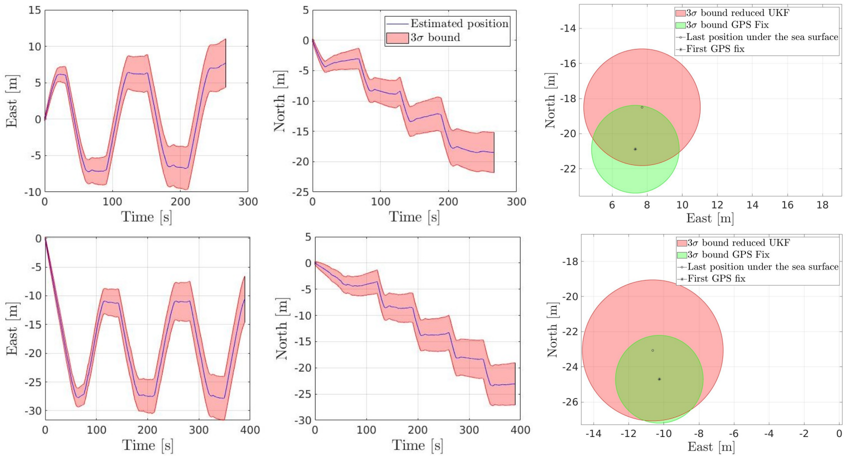


Figure 5.19: On the left and in the center, the position estimation with its 3σ bound obtained with the reduced UKF algorithm respectively along the East and North direction. On the right, the bound of the last position under the sea surface and the first GPS fix measurement with its accuracy bound. Top images refer to mission #1 and bottom images refer to mission #2.

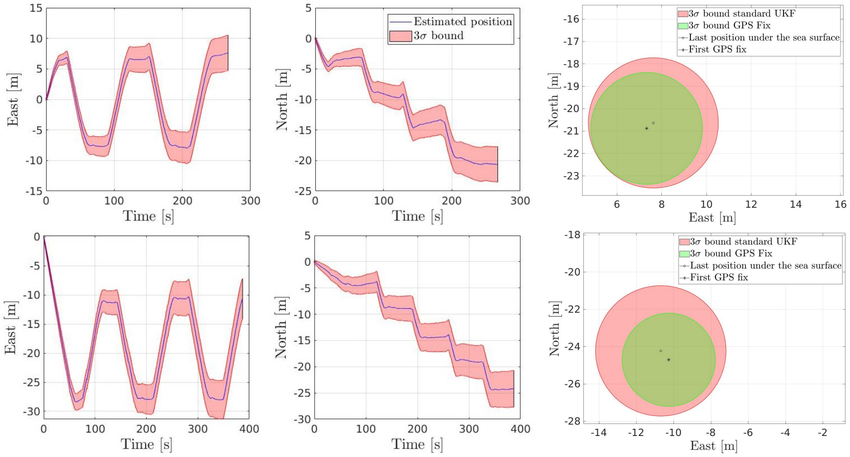


Figure 5.20: On the left and in the center, the position estimation with its 3σ bound obtained with the standard UKF algorithm respectively along the East and North direction. On the right, the bound of the last position under the sea surface and the first GPS fix measurement with its accuracy bound. Top images refer to mission #1 and bottom images refer to mission #2.

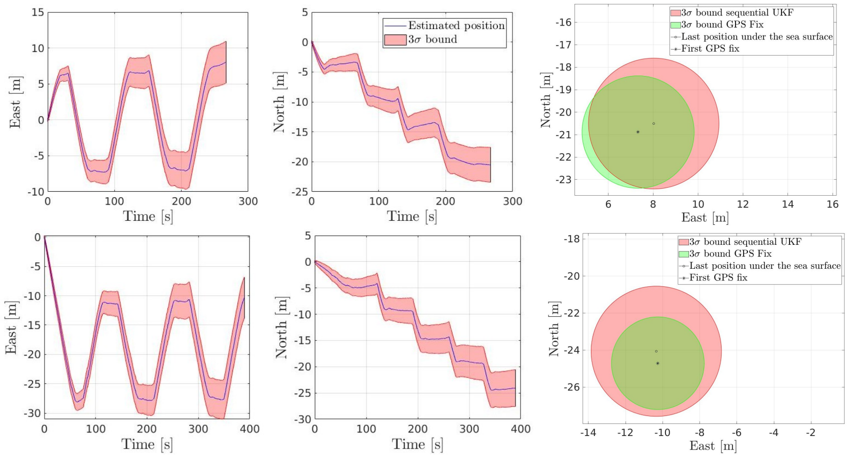


Figure 5.21: On the left and in the center, the position estimation with its 3σ bound obtained with the sequential UKF algorithm respectively along the East and North direction. On the right, the bound of the last position under the sea surface and the first GPS fix measurement with its accuracy bound. Top images refer to mission #1 and bottom images refer to mission #2.

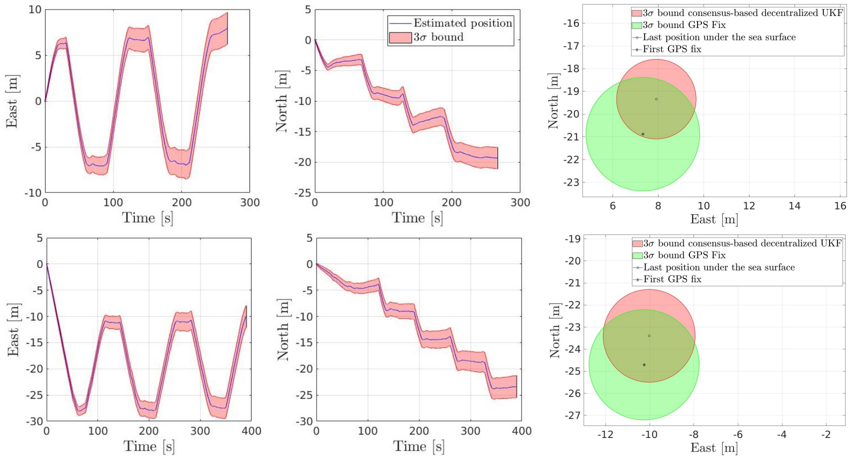


Figure 5.22: On the left and in the center, the position estimation with its 3σ bound obtained with the consensus-based distributed UKF algorithm respectively along the East and North direction. On the right, the bound of the last position under the sea surface and the first GPS fix measurement with its accuracy bound. Top images refer to mission #1 and bottom images refer to mission #2.

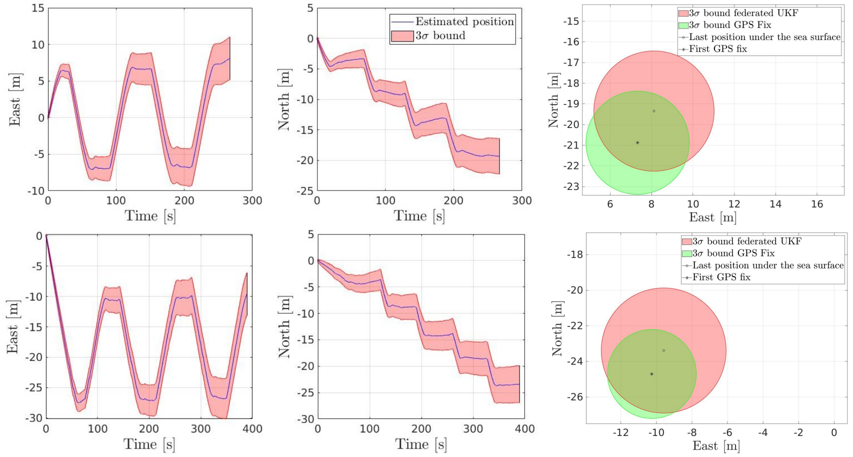


Figure 5.23: On the left and in the center, the position estimation with its 3σ bound obtained with the federated UKF algorithm respectively along the East and North direction. On the right, the bound of the last position under the sea surface and the first GPS fix measurement with its accuracy bound. Top images refer to mission #1 and bottom images refer to mission #2.

Table 5.4: Mean error during mission #1: value at the last waypoint [m].

Navigation strategy	Error ($q = 0.5$)	Error ($q = 0.25$)
Reduced UKF	0.864	2.231
Standard UKF	0.285	0.865
Sequential UKF	0.320	1.118
Consensus-based Distributed UKF	0.254	0.909
Federated UKF	0.168	0.865

Table 5.5: Mean error during mission #2: value at the last waypoint [m].

Navigation strategy	Error ($q = 0.5$)	Error ($q = 0.25$)
Reduced UKF	1.231	1.94
Standard UKF	0.381	0.688
Sequential UKF	0.569	0.831
Consensus-based Distributed UKF	0.692	0.754
Federated UKF	0.637	0.899

reduced dataset has been compared to the one obtained with the full dataset. The employed metrics are defined in Eq. 5.42 and Eq. 5.43.

$$e_i = \|\boldsymbol{\eta}_{1,i}^{RP} - \boldsymbol{\eta}_{1,i}^{TS}\| \tag{5.42}$$

$$\overline{e}_k = \frac{\sum_{i=1}^{k} e_i}{k} \tag{5.43}$$

where $e_i \in \mathbb{R}^+$ represents, at the instant $i \in \mathbb{N}$, the navigation error, $\boldsymbol{\eta}_{1,i}^{RP}$ and $\boldsymbol{\eta}_{1,i}^{TS}$ denote, with respect to the NED reference system, the vehicle position according to the reference path and to the current tested solution, respectively. As introduced before, while the reference path is represented by the estimated trajectory retrieved by applying one of the proposed strategies with a selection test with $q = 1$, the tested path is obtained by estimating the vehicle position with the same strategy of the reference path but with a selection coefficient reduced to $q = 0.5$ or $q = 0.25$. In addition to this, $\overline{e}_k \in \mathbb{R}^+$ denotes the mean of e_i for $i = 1, ..., k$.

Regarding the computational burden, the Central Processing Unit (CPU) burden and the execution time of the filters have been the subject of the analy-

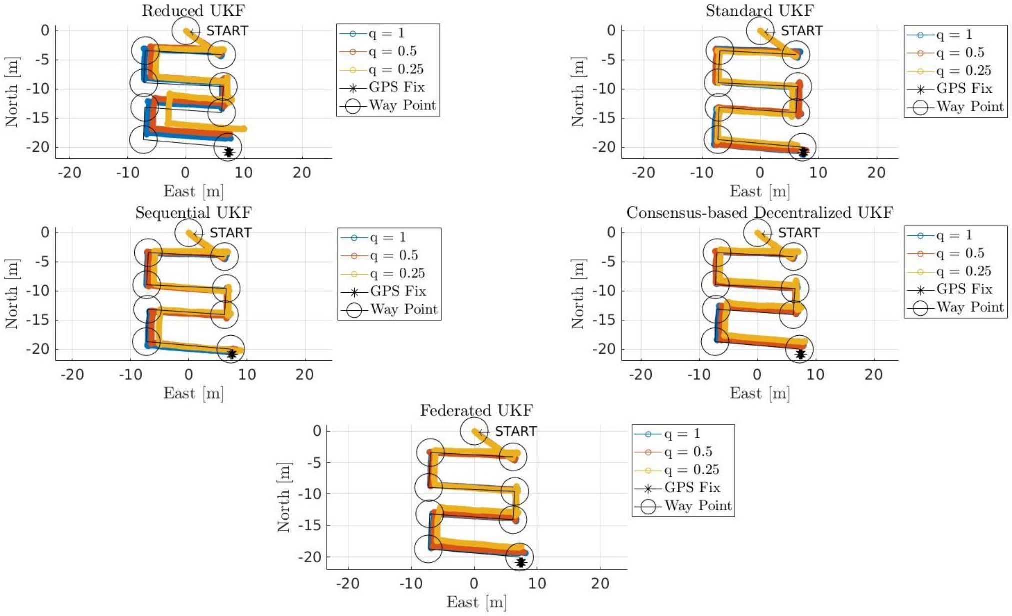


Figure 5.24: Comparison of the estimated trajectories during mission #1 with $q = \{1, 0.5, 0.25\}$.

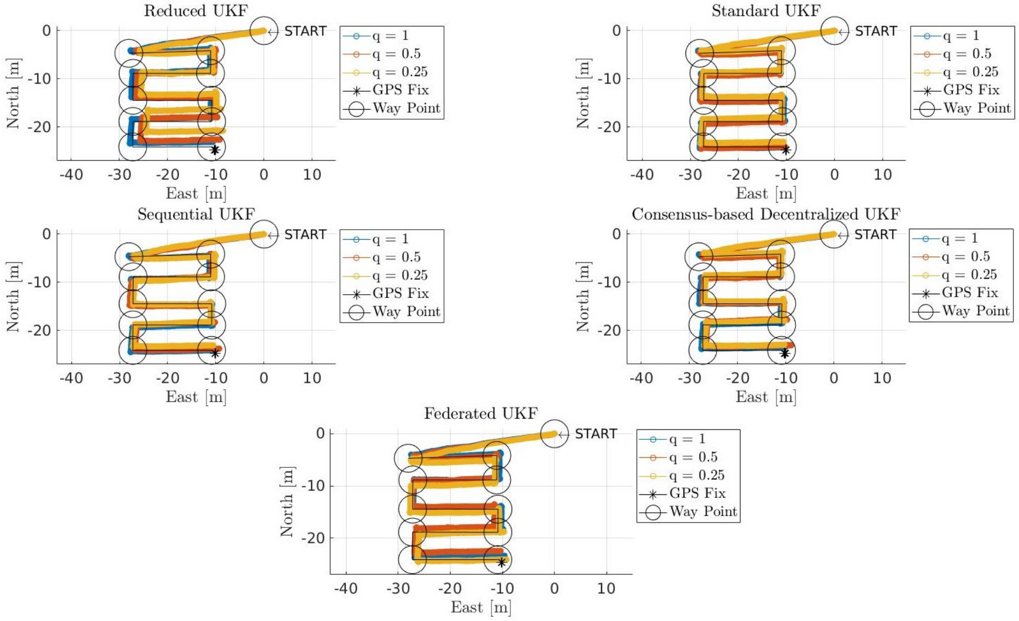


Figure 5.25: Comparison of the estimated trajectories during mission #2 with $q = \{1, 0.5, 0.25\}$.

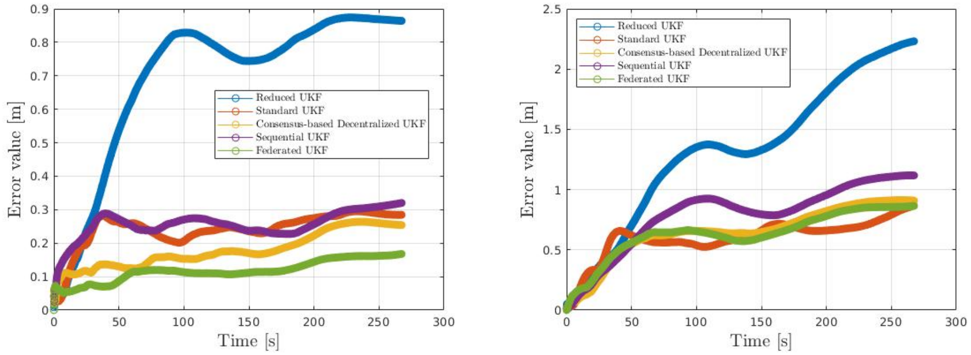


Figure 5.26: Comparison of the mean errors of the proposed strategies during mission #1 with $q = 0.5$ (left image) and with $q = 0.25$ (right image). The error value is computed with Eq. 5.43.

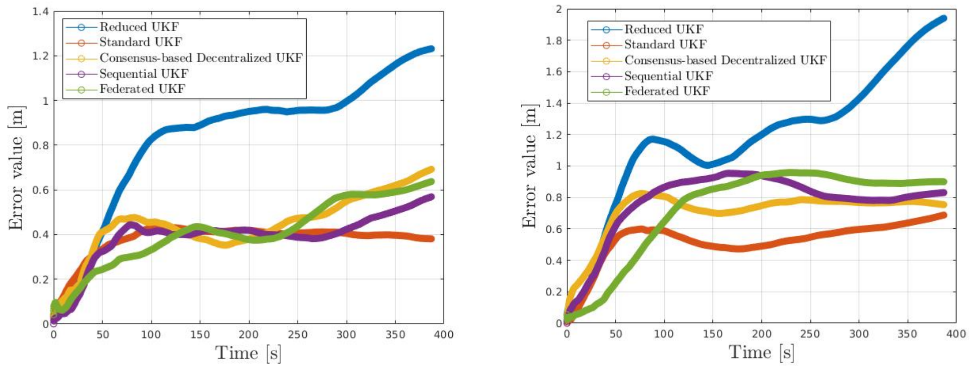


Figure 5.27: Comparison of the mean errors of the proposed strategies during mission #2 with $q = 0.5$ (left image) and with $q = 0.25$ (right image). The error value is computed with Eq. 5.43.

sis. The output of the command *top*, which is a task manager program available in many Unix-like operating systems, has been recorded to store the data. Regarding the execution time, the sum of the requested time to perform both the prediction and the correction steps has been taken into account. The results can be found in Fig. 5.28 for what concern the CPU burden and in Fig. 5.29 for the execution time.

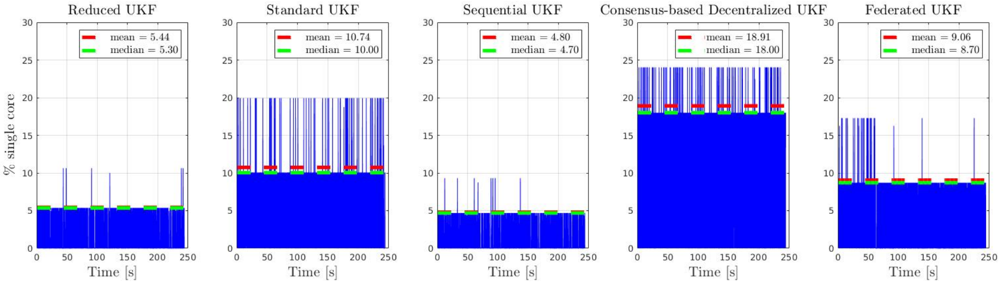


Figure 5.28: CPU burden analysis. In red and green are respectively reported the mean and the median.

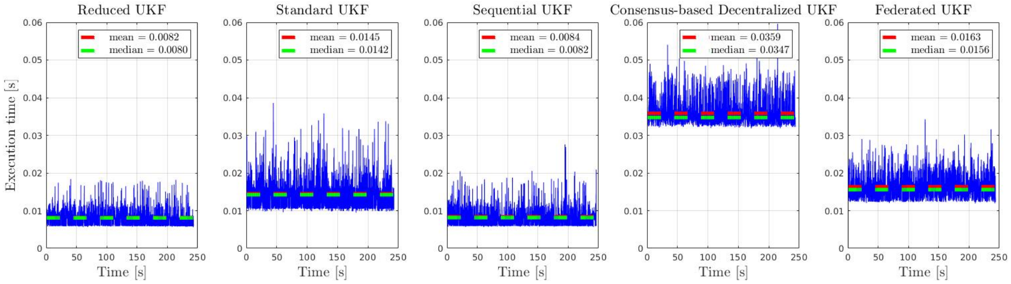


Figure 5.29: Execution time of the filter, calculated at each iteration as sum of the requested time for the prediction and the correction steps. In red and green are respectively reported the mean and the median.

5.6 Analysis of the results

When compared to the reduced UKF, i.e., the filter currently on board Feel-Hippo AUV, the obtained results, due to the employment of a more significant number of speed measurements, show an advantage in terms of navigation estimation quality (see Fig. 5.17, Fig. 5.18, Tab. 5.2, Tab. 5.3). Moreover, the obtained results from the proposed filters are comparable with the standard UKF, which centrally employs all the measurements simultaneously. As a matter of fact, the iterative and parallel fusion strategies guarantee a solid trade-off between localization accuracy and computational cost. More in detail, the sequential UKF strategy provides the minor resurfacing error and the best adherence to both missions' reference trajectories. Furthermore, this strategy guarantees the best adherence to the standard UKF estimated trajectory, which can be considered as the benchmark path. Concerning the federated and the consensus-based decentralized, these strategies guarantee a performance increase with respect to the reduced UKF strategy, thanks to their better exploitation of the available measurements in a framework based on parallel local filtering.

Regarding robustness against the lack of measurements, trivially, the reduction of measurements leads to an increase in navigation error. However, the proposed strategies guarantee better robustness against these issues, providing results comparable with the standard UKF. By observing Fig. 5.26, which are in the presence of 50% and 75% measurement reduction, it can be noticed that both the decentralized and the centralized strategies are similarly insensitive to measurement reduction. Only the reduced UKF, due to its strategy based on the last acquired speed measurement in chronological order, is significantly negatively affected by the measurement reduction (see Tab. 5.4 and Tab. 5.5). Concerning the computational burden, it can be noticed that the sequential UKF has a CPU burden and an execution time similar to the reduced UKF. Still, it can provide a lower resurfacing error comparable with the standard UKF. It is possible to assert that the sequential UKF can represent a solid trade-off between computational complexity and estimation performance. The proposed consensus-based decentralized UKF has the highest CPU burden and execution time due to the particular adopted fusion strategy. Finally, it is necessary to highlight that the federated UKF, despite being a decentralized strategy, has a mean execution time slightly higher than the standard UKF. This last statement highlights the not negligible burden requested by a centralized strategy that, at the same time, processes all the available measurements.

Finally, it is necessary to summarize the pros and cons of each filtering strategy proposed to overcome the limitations of the reduced UKF. The reduced UKF, despite requiring the lowest CPU burden and execution time, provides estimated trajectories whose estimation errors tend to diverge faster than the other filters. Furthermore, this strategy is susceptible to the need for more measurements, which can be limited when they are obtained from payload sensors that acquire data from the surrounding environment. Consequently, the obtained results can be summarized in the following sentences:

- the standard UKF centrally fuses at the same time all the available measurements and it provides in both missions the lowest resurfacing error. It requests a computational burden higher than the other centralized filters, i.e., sequential UKF and reduced UKF, but similar or lower than the decentralized strategies;
- the sequential UKF can be the best solution to guarantee a solid trade-off

between an increase of estimation properties and a reduction of computational load, which can be one of the major limitations in mobile robotics. Reducing the available measurements, the sequential UKF remains much better than the reduced UKF, but it provides the worst results if compared with the other filters.

- the consensus-based decentralized UKF provides the lowest divergence of the estimated covariance when the vehicle is under the sea surface, and no position measurements are available. As illustrated by the Monte Carlo simulations, it is possible to highlight that the filter provides the best results from the consistency point of view and that the position 3σ bound contains the position estimation errors. On the contrary, the employed fusion strategy requires a not negligible execution time and CPU burden, which could limit its applications when the hardware is not appropriate.

- the federated UKF is a decentralized strategy with a requested execution time comparable with the standard UKF. From the results obtained by reducing the measurement availability, the federated UKF is the most insensitive strategy to the lack of measurements.

5.7 Main contributions

Handling redundant observations (e.g., speed information from different sources) still represents an open problem for AUV navigation applications. Indeed, centralized and decentralized fusion strategies have been developed and analyzed (both in terms of localization accuracy and computational burden). In particular, by means of validation with simulated data and real data acquired with an AUV, a sequential (or iterated), a federated, and a consensus-based decentralized technique have been subject of investigation and have been compared with more traditional strategies, as a centralized standard UKF and the UKF proposed in [20] and [68], which has been referred as reduced UKF. In conclusion, the main contributions of this section are:

- the simultaneous employment of different sensing devices for AUV speed measurement, namely a DVL, an optical camera, and an FLS;

- a systematic comparative study among different data fusion techniques (both centralized and decentralized) for AUV navigation in the presence of redundant speed observations;

- a validation of the hereby presented techniques by means of navigation data acquired during autonomous missions performed with an AUV.

Chapter 6

SLAM-based fusion strategies for multisensor navigation

While full smoothers (or batch nonlinear least-squares algorithms) estimate the complete history of poses, fixed-lag smoothers (or sliding window estimators) consider a window of the latest poses, and filtering approaches only estimate the latest state [35]. Both fixed-lag smoothers and filters marginalize older states and absorb the corresponding information in a Gaussian prior. Filtering algorithms enable efficient estimation by restricting the inference process to the latest state of the system. The complexity of the SLAM strategies based on Kalman filtering (e.g., EKF-SLAM) grows quadratically in the number of estimated landmarks [86], [87]. Therefore, a small number of landmarks are typically tracked to allow real-time operation. Full smoothing methods estimate the entire history of the states (vehicle trajectory and 3D landmarks) by solving a large nonlinear optimization problem. Full smoothing guarantees the highest accuracy; however, the real-time operation quickly becomes infeasible as the trajectory and the map grow over time. A breakthrough has been the development of incremental smoothing techniques, which leverage the expressiveness of factor graphs to maintain sparsity and to identify and update only the typically small subset of variables affected by a new measurement.

As for the UKF-based algorithm, the complete pose estimation filter works by following two parallel structures. The first part, employed for attitude esti-

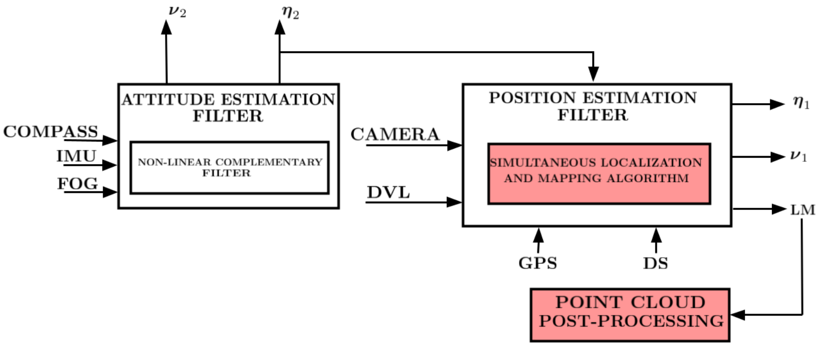

ν_2
η_2
ATTITUDE ESTIMATION FILTER
NON-LINEAR COMPLEMENTARY FILTER
COMPASS
IMU
FOG
CAMERA
DVL
POSITION ESTIMATION FILTER
SIMULTANEOUS LOCALIZATION AND MAPPING ALGORITHM
η_1
ν_1
LM
GPS
DS
POINT CLOUD POST-PROCESSING

Figure 6.1: Framework of the navigation filter, where the modification introduced by the proposed SLAM algorithm, depicted in red, can be noticed in the position estimation filter. Furthermore, the post-processing block, which is described in Appendix A, is reported. The acronym LM stands for landmark and it indicates the landmark positions estimated with the SLAM algorithm.

mation, is used as input in the second part, which is dedicated to position estimation. The SLAM estimator based on DVL and optical measurements is employed as the core of the position estimation filter (see Fig. 6.1).

6.1 Strategies for factor graph constraints computation

6.1.1 DVL-based navigation and mapping

As introduced previously, a DVL-based DR strategy computes the AUV position by integrating the measured linear velocity. Assuming a discrete-time system, the mathematical equation that can be extracted from the previously introduced statement is reported below:

$$\Delta\boldsymbol{\eta}_{k-1,k} = R_b^N(\boldsymbol{\eta}_{2,k-1})\boldsymbol{\nu}_{1,k-1}\Delta T, \tag{6.1}$$

where $\Delta\boldsymbol{\eta}_{k-1,k}$ is the translation between the iteration times $k-1$ and k expressed in the NED reference system, $R_b^N(\boldsymbol{\eta}_2)$ represents the rotation matrix between the NED and the body-fixed frames, $\boldsymbol{\nu}_{1,k-1}$ contains the body-fixed frame linear velocities and ΔT is the fixed sampling time. The output vector of

Eq. 6.1 represents the first three components of $m_{XYZ-Y}(x_{i-1}, x_i)$ defined in Eq. 3.85.
While the primary application field of a DVL is vehicle navigation, it can also be employed to provide an approximated sea floor bathymetry. Indeed, a DVL has four acoustic beams, each pointing in a different direction, which can be employed to acquire the 3D location of 4 points of the sea bottom during each speed measurement. One beam of the DVL is represented by the vector $\boldsymbol{P}$ and goes from the DVL to the sea floor:

$$\boldsymbol{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = Z \begin{bmatrix} \tan\gamma\cos\beta \\ \tan\gamma\sin\beta \\ 1 \end{bmatrix} \tag{6.2}$$

where γ is the angle of the beam from the DVL z-axis, β is the angle from the x-axis on the xy-plane, Z is the vertical component of $\boldsymbol{P}$ (see Fig. 6.2). It is worth noting that $\boldsymbol{P}$ (whose module is the output of the DVL) is expressed in terms of Z and the constant angles β and γ. The points located thanks to the DVL beams cannot be employed as additional constraints in the navigation pose graph because they do not link any node of the graph. Still, they can easily be used to increase the number of points in the estimated map of the sea bottom. Indeed, by knowing the vehicle's actual position from the navigation algorithm, the location of the four beams can be converted from the DVL frame to the NED reference system.

6.1.2 Mono visual SLAM

The visual SLAM algorithm employed in the developed navigation framework is a feature-based monocular SLAM system that operates to estimate the camera trajectory and an environment map. The basic idea of the SLAM system introduced in the navigation filter takes inspiration from the algorithms proposed in [45], [46]. Following the results reported in [76], [77], where accurate comparisons between several feature detectors are explained, ORB feature detector has been chosen as the preferable solution instead of Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) and Accelerated-KAZE (AKAZE). The SLAM framework works following two main processes: tracking and local mapping. While the tracking is dedicated to locating the camera by using every frame ad to decide when to insert a new keyframe, the local mapping process employs the new keyframes to perform a local bundle adjustment optimization to compute a reconstruction of the surrounding envi-

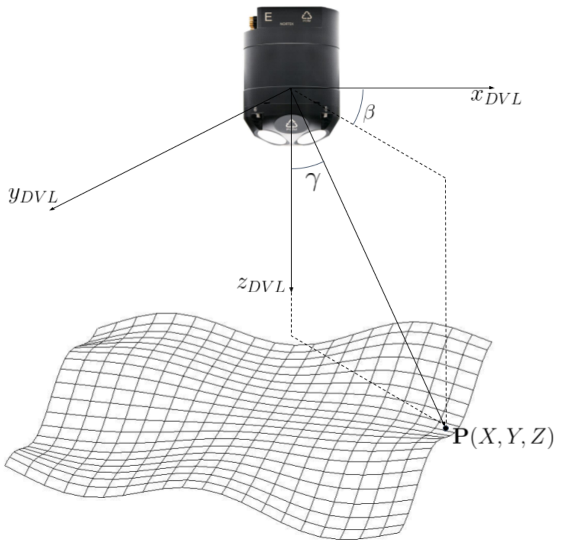


Figure 6.2: The DVL beam vector components.

ronment.

The map initialization process, dedicated to the computation of the relative pose between two frames and the triangulation of an initial set of map points, is executed by exploiting the method reported in [45]. The initialization is performed by employing only the data from the monocular camera; consequently, the retrieved results are up to a scale factor. The scale factor estimation uses additional onboard sensors and will be described in Section 6.2.2. The initialization procedure computes two geometrical models in parallel: a homography matrix, where a planar scene hypothesis is taken into account, and a fundamental matrix, which is supposed to have a non-planar scene. The algorithm initializes the system when the estimated configuration is safe and a low number of matches or unstable correspondences does not corrupt the map. The ORB features are extracted from the current and the reference frames and are matched. If enough matches are found, the homography and the fundamental matrix are computed in parallel by exploiting the Direct Linear Transformation (DLT) and the 8-points algorithms, respectively. The RAndom SAmple Consensus (RANSAC) outlier removal procedure has been adopted in both cases. Two score values S_H and S_F have been computed for the homography and the fundamental matrices, respectively, by employing the formulas reported in [45]. It is necessary to underline that if the scene is planar or with a low parallax, it can be correctly described with a homography and the retrieved fundamental matrix is obtained from a not well-constrained problem, leading to erroneous results. The coefficient

$$R_H = \frac{S_H}{S_H + S_F} \tag{6.3}$$

is employed to select between homography and fundamental matrices. The homography is selected if $R_H > 0.45$, otherwise, the fundamental matrix is chosen. The selected threshold value has been chosen empirically. Once a model is selected, the associated motion hypothesis is retrieved, the map points are triangulated, and a full bundle adjustment is performed.
The tracking algorithm is applied to every frame captured by the camera. ORB features are extracted from each image, and a matching algorithm is employed to retrieve map points in new images and to determine the camera motion. After this step, the local map can be projected into the frame to increase the number of point correspondences. Only the keyframes are inserted in the overall factor graph to maintain bounded the global system complexity. To select a new keyframe, three conditions have to be satisfied:

- more than 20 frames have been elaborated from the last keyframe insertion;
- current frame tracks at least 50 points;
- current frame tracks less than 90% points than the last keyframe.

The parameter values have been chosen following the inputs provided by the authors of [45]. When a new keyframe is selected, the covisibility graph is updated. This step is accomplished by adding a new node and updating the edges resulting from the shared map points with the last inserted keyframes. To be passed to the global map, map points must be visible over three keyframes to guarantee that they are correctly tracked and triangulation can be accurately performed. Finally, the map points and the pose associated with the keyframe are passed to the global navigation factor graph to perform the optimization process.

6.2 Factor graph framework development

6.2.1 Overall factor graph

The information from the available onboard sensors has been encoded as measurement factors to constrain the optimization, whose solution represents the

MAP estimate. Inspired by [41], the following factors have been included:

- a relative 4D pose-to-pose constraint on x, y, and z translation and yaw rotation, thanks to the measurements coming from the DVL and the yaw estimated by the attitude estimator;
- a unary 2D constraint on pitch and roll rotations, obtained from the attitude estimation filter;
- a unary 1D constraint on z translation thanks to the DS measurements;
- a unary constraint on x and y translation exploiting GPS observations;
- a relative 6D pose-to-pose constraint on x, y, and z translation and roll, pitch, and yaw rotation, thanks to the relative pose estimated through the monocular camera and properly scaled;
- a camera-based landmark constraint on the vehicle pose and the landmark position for each feature seen with the monocular camera over multiple images.

The implemented approach adds a new state only when at least one observation from GPS, DVL, DS, or when the visibility is acceptable, the camera is available. The link between adjacent nodes is maintained by collapsing the relative motion XYZ-Y in a single compound constraint, where simple DR is performed between the two consecutive nodes with the last acquired DVL measurements. As introduced in Section 3.3.3, the pose T_{x_i} can be represented with a vector $\begin{bmatrix} X_{x_i} & Y_{x_i} & Z_{x_i} & \phi_{x_i} & \theta_{x_i} & \psi_{x_i} \end{bmatrix} \in \mathbb{R}^6$ that encodes the state at the generic instant. Mathematically, at time k, the optimization problem can be written as

$$\begin{aligned}
\mathcal{X}_k^* = \underset{X}{\operatorname{argmax}} \;\; & \textstyle\sum_{i=1}^{k-1} \left(\| m_{XYZ-Y}(x_{i-1}, x_i) \ominus o_{i-1,i} \|^2_{\Sigma_{o_{i-1,i}}} + \| m_{RP}(x_i) \ominus r_i \|^2_{\Sigma_{r_i}} \right) + \\
& + \textstyle\sum_{i \in \mathcal{Z}} \| m_Z(x_i) - z_i \|^2_{\Sigma_{z_i}} + \\
& + \textstyle\sum_{i \in \mathcal{G}} \| m_{XY}(x_i) - \boldsymbol{g}_i \|^2_{\Sigma_{g_i}} + \\
& + \textstyle\sum_{i,j \in \mathcal{C}} \| m_{XYZ-RPY}(x_i, x_j) \ominus p_{i,j} \|^2_{\Sigma_{p_{i,j}}} + \\
& + \textstyle\sum_{j \in \mathcal{LM}, i \in \mathcal{C}} \rho \left(\| \boldsymbol{p}_{ij} - \pi_i(T_{x_i} \boldsymbol{P}_j) \|^2_{\Sigma_{lm_i}} \right) + \\
& + \| T_{x_0} \ominus T_{x_{prior}} \|^2_{\Sigma_{lm_i}}
\end{aligned} \tag{6.4}$$

$\{m_{XYZ-Y}(\cdot), o_{i-1,i}, \Sigma_{o_{i-1,i}}\}$, $\{m_{RP}(\cdot), r_i, \Sigma_{r_i}\}$, $\{m_Z(\cdot), z_i, \Sigma_{z_i}\}$, $\{m_{XY}(\cdot), \boldsymbol{g}_i, \Sigma_{g_i}\}$, $\{m_{XYZ-RPY}(\cdot), p_{i,j}, \Sigma_{o_{i,j}}\}$ are the measurement functions, the measured values and covariances defined in Section 3.3.3 and associated to the previously

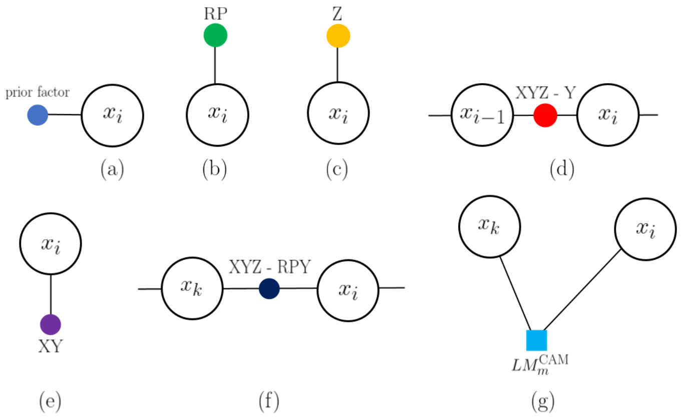


Figure 6.3: Graphical representation of the constraints employed in the factor graph for the developed navigation strategy. It is possible to notice the prior factor (a), the RP unary constraint (b), the Z unary constraint (c), the XYZ-Y pose-to-pose constraint (d), the XY unary constraint (e), the XYZ-RPY pose-to-pose constraint (f), the visual landmark constraint (g).

introduced factors. While $\mathcal{Z}$, $\mathcal{G}$ and $\mathcal{C}$ are the set of pose nodes for which DS, GPS and camera measurements respectively occur, $\mathcal{LM}$ is the set of landmark nodes. $T_{x_{prior}}$ is the prior constraint on the first pose, which is necessary to anchor the state evolution to a global coordinate frame (Fig. 6.4).

In terms of implementation, the GTSAM library [88] has been used as the back-end for the localization solution. Further information can be found in [38], [39]. iSAM2, which is the latest evolution of the incremental smoothing and mapping solution developed in GTSAM, allows only the typical small subset of variables affected by a new measurement, i.e., the measurement function and associated covariances, to be identified and updated, thus limiting the computational load of the estimation, offering a trade-off between accuracy and efficiency. Several issues affect the vision in underwater environments, which can negatively influence the employment of visual SLAM algorithms. Specifically, while scattering reduces light intensity causing a loss of contrast and haze in underwater images, light absorption leads to a decrease in the color quality of underwater images. Light attenuation in water introduces degradation in underwater images, such as poor colors, decreased contrast, haziness, and blurring, making them hardly usable for the filter. Thus it is necessary to guarantee that

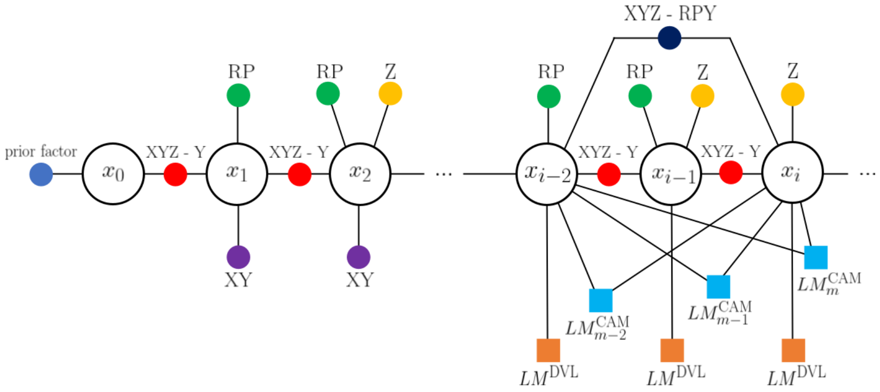


Figure 6.4: Example of the factor graph at the iteration i constrained with vision-based landmarks and all the onboard sensors.

the visual part of the navigation framework, which is dependent on uncontrollable external conditions, can be correctly inserted or removed from the factor graph. Only when the visual SLAM algorithm is correctly initialized and the current scale factor is computed, it is possible to insert keyframe poses and map point locations in the factor graph. Edges computed thanks to other onboard sensors, which do not suffer from visibility limitations, are inserted in the whole factor graph during the entire mission. The developed system, through the map points obtained from the vision system and the DVL beams, can build a map of the surrounding environment independently from the visibility. Indeed, when the reduced visibility impedes the usage of the visual-based features as map points, DVL-based beams can be employed to build an approximated map of the sea bottom. The quality and resolution of the produced map depend on the availability of the visual landmarks. Still, thanks to the DVL measurements, the reconstruction can be performed for the whole mission. Considering that, when an AUV accomplishes an underwater mission, the sea bottom texture can change very fast, and its depth can increase rapidly, the possibility to guarantee reconstruction of the surrounding environment, even approximated, represents a helpful advantage. Obviously, it is necessary to highlight that the DVL beams cannot be employed as landmark nodes in the factor graph. Still, they can only be added to the map utilizing the sensor geometrical model. It is necessary to highlight that underwater SLAM fusing camera, and DVL sensors can increase the localization accuracy and robustness thanks to the excellent complement between these two sensors: DVL provides reliable motion estimates for underwater visual SLAM, extending SLAM's robustness and operation even without visual features, and vision, when applicable, helps the estimation process by introducing visual landmarks which increase the constraints on the vehicle position.

6.2.2 Scale factor ambiguity resolution

This procedure, which is executed every time the visual SLAM algorithm is correctly initialized, has two main purposes, the scale factor ambiguity resolution and accurate compensation of the fixed rototranslation between the camera and the body frames. This transformation is represented as a similarity transformation composed of a scale factor s, a translation vector $\boldsymbol{t}_{c,b} = \left[\begin{array}{ccc} t^x_{c,b} & t^y_{c,b} & t^z_{c,b} \end{array}\right]^\top$ and a rotation matrix $R^b_c = R_z(\psi^b_c)R_y(\theta^b_c)R_x(\phi^b_c)$. It is based on comparing the trajectories estimated through the DVL and the other inertial sensors and the camera. It is necessary to notice that until the scale factor has not been estimated, the measurements obtained thanks to the visual SLAM algorithm are not inserted in the whole factor graph. Considering this algorithm's two purposes and that, usually, underwater vehicles for survey missions execute planar trajectories at constant depth, the problem has been solved with a two-step algorithm. In particular, while the first part of the algorithm determines a closed-form solution for the x and y directions, yaw rotation, and the scale factor, the second part optimizes the whole scaled rototranslation with an iterative algorithm. This framework has been adopted due to the limitations introduced by the particular motion executed by the AUV. Indeed, on the one hand, the optimal closed-form solution estimated with 3D points that almost lie on a plane cannot correctly estimate the roll and pitch angles of the rigid transformation between the two considered reference frames. On the other hand, the iterative algorithm locally converges and requires an initial guess in the neighborhood of the exact solution, which can be measured directly on the vehicle or evaluated through the closed-form solution.

The two steps of the algorithm are described in detail. Firstly, the closed-form solution is found by computing the trajectory alignment transformation with translational component on the xy-plane of the trajectory estimated with the DVL and the camera and with rotational component computed with respect to the perpendicular axis to this plane. Given the DVL-based positions $\{\boldsymbol{p}^{DVL}_i\}^N_{i=1}$ and the camera-based positions $\{\boldsymbol{p}^{CAM}_i\}^N_{i=1}$, it is necessary to determine the optimal similarity transformation $S^* = \{s^*, R^{b\,*}_c, \boldsymbol{t}^*_{c,b}\} = \{s^*, \psi^{b\,*}_c, t^{x\,*}_{c,b}, t^{y\,*}_{c,b}\}$ that satisfies the minimization problem reported in Eq. 6.5.

$$S^* = \underset{s, R_c^b, \boldsymbol{t}_{c,b}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\boldsymbol{p}_i^{DVL} - sR_c^b \boldsymbol{p}_i^{CAM} - \boldsymbol{t}_{c,b}\|^2 \tag{6.5}$$

where it is necessary to suppose that

$$R_c^b = \begin{bmatrix} c_{\psi_c^b} & -s_{\psi_c^b} & 0 \\ s_{\psi_c^b} & c_{\psi_c^b} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.6}$$

$$\boldsymbol{t}_{c,b} = \begin{bmatrix} t_{c,b}^x \\ t_{c,b}^y \\ 0 \end{bmatrix} \tag{6.7}$$

The solution of this least squares problem can be found using the method explained in [89].

The second step works with Ceres Solver, an open-source library that provides a rich set of tools to construct and solve an optimization problem. Ceres solves robustified bounds constrained non-linear least squares problems of the form:

$$\begin{array}{ll} \min\limits_{\boldsymbol{x}} & \frac{1}{2} \sum\limits_i \rho_i \left(\|f_i(x_{i_1}, ..., x_{i_k})\|^2\right). \\ l_j \le x_j \le u_j & \end{array} \tag{6.8}$$

The expression $\rho_i\left(\|f_i(x_{i_1}, ..., x_{i_k})\|^2\right)$ represents the residual block, where $\rho_i(\cdot)$ is the loss function used to reduce the influence of outliers on the solution and $f_i(\cdot)$ is the cost function that depends on the parameters block $\{x_{i_1}, ..., x_{i_k}\}$. l_j and u_j are the lower and upper bounds on the parameter block x_j.

Defining the state $\boldsymbol{x} = \begin{bmatrix} s & \phi_c^b & \theta_c^b & \psi_c^b & t_{c,b}^x & t_{c,b}^y & t_{c,b}^z \end{bmatrix}^\top$, the loss function is assumed to be the identity function, the cost function is the same as in the first step of the algorithm

$$f(\boldsymbol{x}) = \boldsymbol{p}_i^{DVL} - sR_c^b \boldsymbol{p}_i^{CAM} - \boldsymbol{t}_{c,b} \tag{6.9}$$

where, unlike the previous case, it is supposed that

$$R_c^b = \begin{bmatrix} c_{\theta_c^b} c_{\psi_c^b} & s_{\phi_c^b} s_{\theta_c^b} c_{\psi_c^b} - c_{\phi_c^b} s_{\psi_c^b} & c_{\phi_c^b} s_{\theta_c^b} c_{\psi_c^b} + s_{\phi_c^b} c_{\psi_c^b} \\ c_{\theta_c^b} s_{\psi_c^b} & s_{\phi_c^b} s_{\theta_c^b} s_{\psi_c^b} + c_{\phi_c^b} c_{\psi_c^b} & c_{\phi_c^b} s_{\theta_c^b} s_{\psi_c^b} - s_{\phi_c^b} c_{\psi_c^b} \\ -s_{\theta_c^b} & s_{\phi_c^b} c_{\theta_c^b} & c_{\phi_c^b} c_{\theta_c^b} \end{bmatrix} \tag{6.10}$$

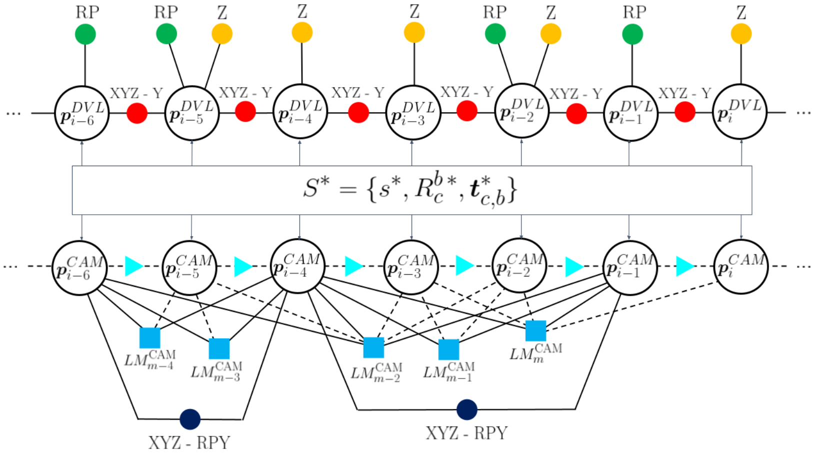


Figure 6.5: Comparison of the two factor graphs (e.g., the DVL-based graph on the top and the camera-based graph on the bottom of the image) employed for the scale factor ambiguity resolution. The dashed lines in the bottom graph are the edges which are not reported in the whole graph. For ease of reading, a one-to-one association between the two graphs is considered.

$$\boldsymbol{t}_{c,b} = \begin{bmatrix} t_{c,b}^x \\ t_{c,b}^y \\ t_{c,b}^z \end{bmatrix}. \tag{6.11}$$

The initial guess and the upper and lower bounds are computed thanks to the values estimated in the closed-form solution. Considering that this is a small problem with few parameters and relatively dense Jacobians, dense QR factorization is the method of choice [90].

6.2.3 Reset procedures

Although iSAM2 reduces the variables to be optimized to a small subset, it is necessary to apply a reset procedure to maintain a limited factor graph and avoid increasing nodes and edges. In particular, considering that the presence of visual landmark nodes constrains several pose nodes, the computational burden tends to increase at every iteration step, and the factor graph is more arduous to be managed. Two factor graph reset procedures have been developed to avoid the increase of the graph size, where the first is dedicated to compacting the factor graph without reducing the visual landmark nodes, and the second operates on

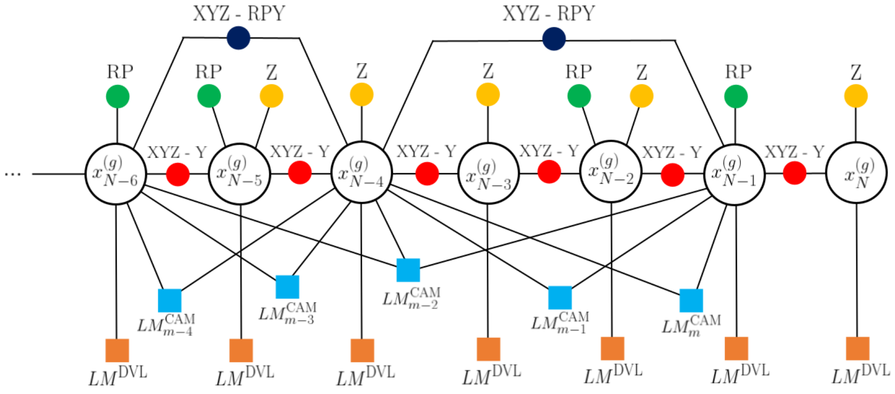


Figure 6.6: Last nodes of the factor graph g constrained with vision-based landmarks and all the onboard sensors.

the whole factor graph reducing all the information to the ones contained in the last node. While the first reset strategy will be called keyframe reset, the second one will be referred as global reset. One of the two reset strategies is applied when the number of pose nodes of the factor graph reaches a value equal to N. The status of the factor graph is checked to decide which one of the two strategies are applied. In particular, the keyframe reset procedure is recalled only if the visual SLAM algorithm is active and for a maximum number of consecutive times equal to p. The last condition is set to maintain control of the increase of the execution time of each filter iteration. When the visual SLAM part of the navigation algorithm is not working due to the external visibility conditions or when the factor graph is reset for the $(p+1)$-th time, the global reset algorithm is employed. It is necessary to notice that the keyframe reset procedure does not delete all the information contained in the previous pose graph. Still, only the ones related to the IMU, DVL, and DS measurements are removed. Indeed, this information is compressed in a new framework, which contains all the properties to be transferred from the previous to the following factor graph. On the contrary, the global factor reset reduces all the information to be transferred to the new factor graph to the ones in the last node of the previous factor graph.

Both the reset strategies are now analyzed in detail to outline which information is passed from the previous to the actual graph and how these measurements are compressed in the new framework. Considering the keyframe reset procedure and referring to Fig. 6.6 and Fig. 6.7, the following actions are performed to obtain the graph $g+1$ from the graph g.

- The $i+1$ keyframe pose nodes are transferred from the previous to the actual factor graph. The first keyframe node, as the one associated with the state $x_k^{(g)}$, is constrained with a prior factor with the last estimated value. All the subsequent $i-1$ keyframe nodes are determined by an XYZ-RPY factor obtained from each last estimated value and the associated covariance.

- All the $m+1$ visual landmark points are transferred from the previous to the actual factor graph. They are employed to maintain constraints between all the keyframe pose nodes. Each landmark node is reported in the current graph with its last estimate and covariance and all the vision-based edges.

- The last pose node associated with the state $x_N^{(g)}$, even if it is not a keyframe node, is transferred to the actual graph to be employed as starting point to insert the acquired measurements as constraints. This node is constrained to the last keyframe node with an XYZ-RPY odometry factor computed from the last pose estimated values of the two nodes. The relative rototranslation transformation is thus computed and applied as a constraint.

All the DVL-based landmarks are reported in the global NED reference frame using the poses estimated with the graph g, and they are employed to build the point cloud for the seabed reconstruction. Even though the whole graph has been reset, the visual SLAM part, if the visibility is acceptable, continues to compute poses and visual landmarks, which are inserted in the new graph and connected to the keyframe nodes passed from the previous graph. Furthermore, until a new keyframe is not computed, the new nodes are inserted thanks to the DVL-based DR, the DS measurements, and the attitude estimator filter outputs.

Considering the global reset procedure and referring to Fig. 6.6 and Fig. 6.8, the following actions are performed to obtain the graph $g+1$ from the graph g.

- Only the last pose node associated with the state $x_N^{(g)}$ is transferred to the actual graph to be employed as starting point to insert the acquired measurements as constraints. It is constrained with a prior factor with the last estimated value.

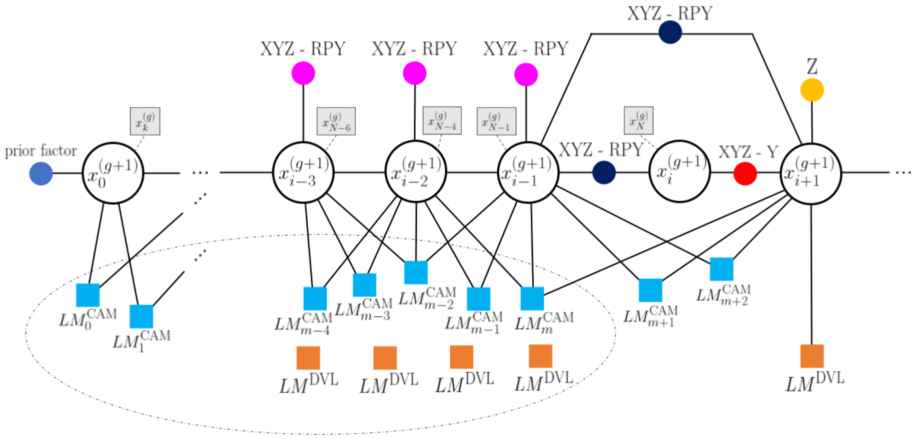


Figure 6.7: First nodes of the factor graph $g+1$ after the employment of the keyframe reset procedure. The values in the grey boxes represent the corresponding states taken from the previous factor graph g and transferred to the actual graph $g+1$.

- The visual landmarks and the keyframe poses are not transferred from the previous to the actual graph. All positions of the estimated DVL-based and visual landmarks are saved as estimated in the last optimization of the previous graph, and they are employed to build the point cloud for the seabed reconstruction.

Even if the visibility is acceptable, the visual SLAM algorithm is reinitialized, the scale factor is again computed, and no information is transferred from the vision-based part of the previous graph. Despite the loss of some helpful information, the global reset procedure is necessary to limit the algorithm's computation burden.

6.3 Validation in simulated environment

To validate the developed DVL and camera-based SLAM algorithm, realistic simulations were performed by means of the Unmanned Underwater Vehicle Simulator (UUV Simulator). In particular, while navigation performance has been evaluated thanks to a Monte Carlo simulation similar to the one described in Chapter 5, mapping capabilities have been analyzed with a lawnmower survey

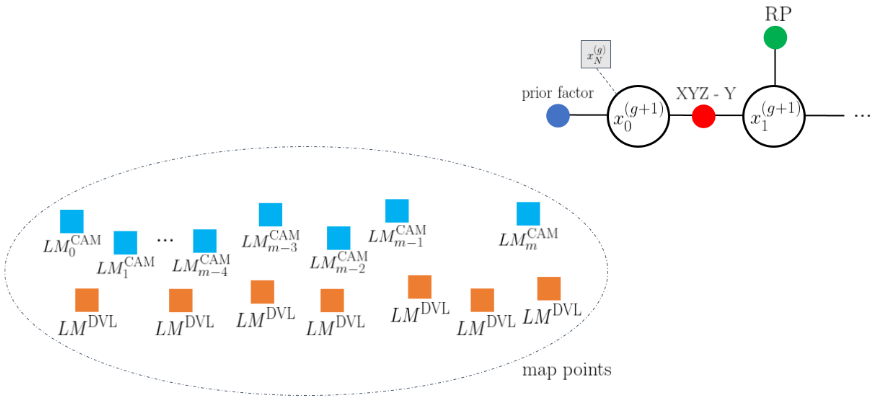


Figure 6.8: First nodes of the factor graph $g+1$ after the employment of the global reset procedure. The values in the grey boxes represent the corresponding states taken from the previous factor graph g and transferred to the actual graph $g+1$.

at a constant depth over a simulated seabed generated with a known mathematical function $z = f(x, y)$. The obtained results have been employed to evaluate the goodness of the whole algorithm and some of its main features, such as the reset procedure and the scale factor computation algorithm. To focus attention on the navigation and mapping capabilities of the filter, the DVL and the camera have been modeled thanks to the simulator features. The realistic simulations were based on the dynamic model of FeelHippo AUV implemented in the UUV Simulator and on modeling all the onboard sensors. In particular, the DVL beams have been modeled by applying a noise in the measured value, which determines a noise in the measured velocity. The camera has been modeled with a noise in the pixel position of the acquired image, which influences both the vehicle and landmark position estimation.

During the Monte Carlo simulations, the position filter was fed with the data coming from the simulated sensors, as the GPS, when the vehicle was higher than a fixed depth, depth sensor, DVL and camera. To increase adherence to the real dataset, the DVL speed measurements have been published with a 5 Hz rate, and the camera acquired images with a frequency of 10 Hz. The proposed strategies have been tested on a vehicle whose dynamic behavior has been simulated using the model implemented in UUV Simulator, which has traveled a rectangular path at a fixed depth of 2 m. A Monte Carlo simulation with 100 iterations has been performed. The position errors and the estimated

3σ bounds along the East and North directions are reported in Fig. 6.9. The covariance trend follows the trajectory described by the vehicle. Still, globally it is possible to highlight an increase in performance in terms of divergence with respect to the UKF filters. Indeed, while in the UKF-based filters the 3σ bounds are characterized by a circular shape, the SLAM algorithm, due to the presence of visual landmarks which constrains the vehicle position, provides an elliptic 3σ bound with major axis perpendicular to the direction followed by the vehicle. Despite its particular shape, the 3σ bound continuously diverges when the vehicle is under the sea surface, and no position measurements are available, correctly representing the behavior of the AUV.
Furthermore, as in the previous section, the estimated resurfacing position has been compared with the theoretical first GPS fix and its 3σ bound (Fig. 6.10). The resurfacing positions estimated in all the Monte Carlo simulations fall inside the 3σ bound, guaranteeing reasonable estimations. Furthermore, it is possible to compare the 3σ bound estimation obtained from the filter and the 3σ bound estimation obtained from the simulated data, evaluating the latter by computing the best normal distribution approximating the estimated resurfacing positions with respect to the theoretical ones (Fig. 6.11).

Analyzing the results obtained from the lawnmower survey at a constant depth of 5 meters and comparing the estimated trajectory with the ground truth provided by the simulator, it is possible to notice that the divergence over time of the navigation error is reduced (see Fig. 6.12). Indeed, even if a global loop closure on the visual keyframes is not performed, the presence of the highly accurate DVL measurements can maintain a low estimation error drift. Furthermore, Fig. 6.12 shows the estimated trajectory on the NED frame, where it is possible to notice the points where the system has been reset. Considering that the simulated seabed has been textured with a feature-rich image, it is necessary to see that the visual part of the SLAM algorithm continues to work for the whole trajectory. Thus, both reset strategies have been employed to limit the computational burden. Fig. 6.13 reports the estimated trajectory and the generated point cloud. It is possible to evaluate the algorithm mapping capabilities by comparing the estimated point cloud and the function employed to simulate the seabed. Considering that several outliers are kept in the point cloud during the SLAM algorithm, which negatively influences the seabed reconstruction, the estimated landmarks are elaborated to eliminate the wrong points and to downsample the cloud. Consequently, the seabed reconstruction capabilities of the developed algorithm are analyzed in Appendix A, where the employed post-processing strategies are described.

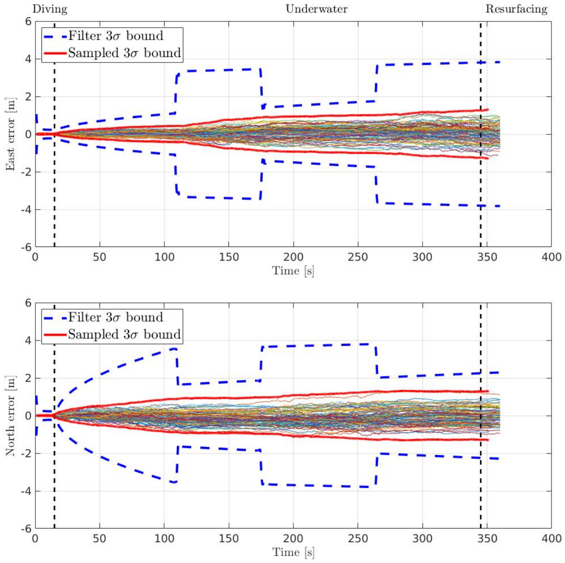


Figure 6.9: East and North position estimation errors versus their 3σ bounds obtained from 100 simulation analysis with the SLAM algorithm. The σ values are computed as the square-root of the corresponding diagonal element of the estimated covariance matrix.

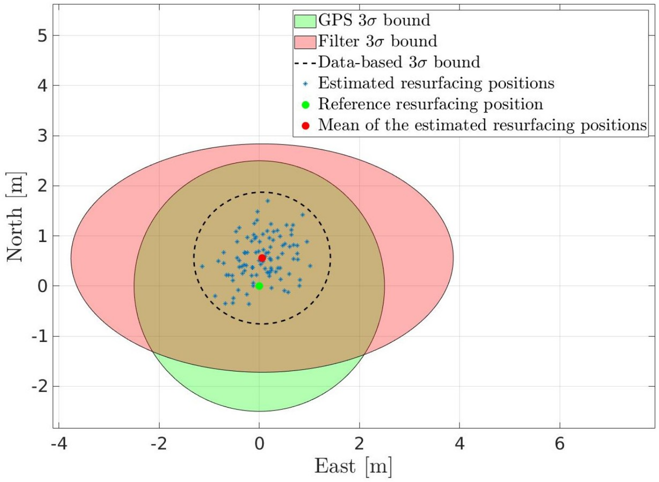


Figure 6.10: The estimated resurfacing positions versus the theoretical GPS fix position obtained from 100 simulation analysis for the SLAM algorithm.

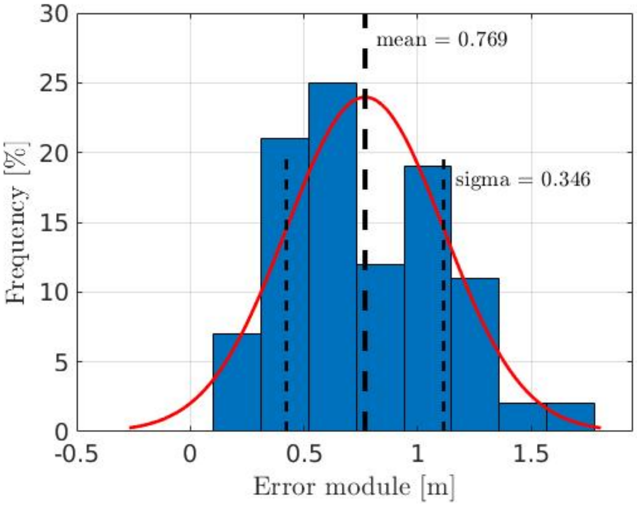


Figure 6.11: Histograms containing the estimated resurfacing position errors obtained from 100 simulation analysis for the SLAM algorithm.

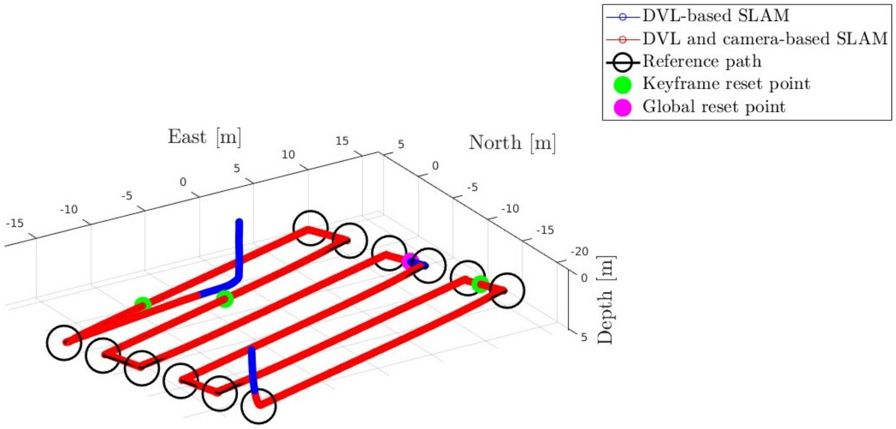


Figure 6.12: 3D plot of the estimated trajectory in the NED reference system, where the reset points and the areas where vision is not working are highlighted.

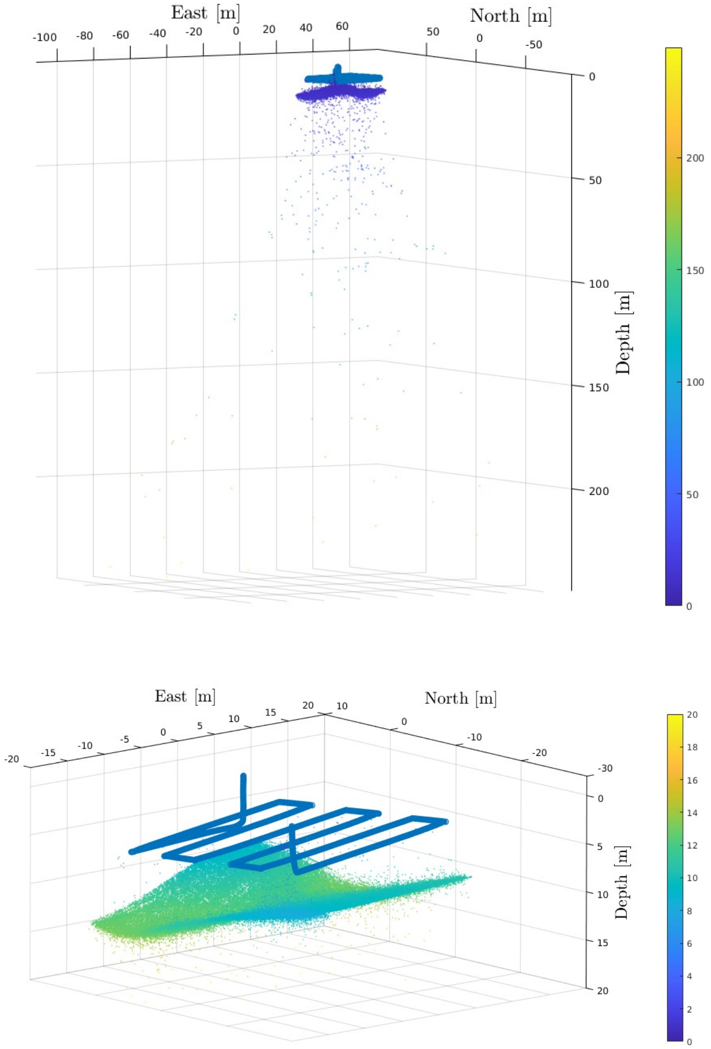


Figure 6.13: Representation of the point cloud and the travelled trajectory estimated through the SLAM algorithm. While on the top image the entire point cloud is reported and, due to the presence of outliers, the depth scale is too extended, on the bottom image a zoom on the region of interest is performed.

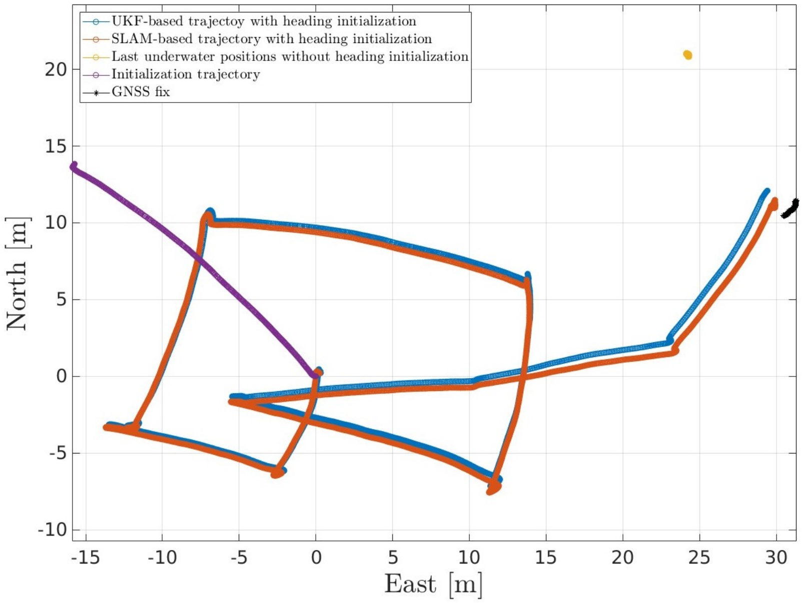


Figure 6.14: Comparison between the trajectories estimated with the SLAM algorithm and the UKF algorithm during the mission accomplished in Stromboli Island, Messina (Italy). A ground truth, when the vehicle was under the sea surface, is not available, but the first GNSS fix when the vehicle resurfaces can be employed as reference to evaluate the resurfacing error.

6.4 Experimental results

The presented navigation and mapping strategy has been tested and validated by employing experimental data recorded in Stromboli Island, Messina (Italy), in September 2022, during an autonomous underwater mission performed in the framework of the project PATHFinder. During its autonomous navigation along a pre-programmed path, the payload sensors were switched on, and the vehicle acquired both acoustic and optical data. GNSS readings obtained from the satellites of the Galileo system were collected before FeelHippo AUV dove and after it resurfaced. They have been employed as ground truth to compute the resurfacing error and to globally reference the trajectory and the map.

The developed SLAM strategy has been compared with the UKF algorithm proposed in Section 5.3.2, which has been employed as a reference filter for the previous analysis. In particular, to ensure uniformity of the employed sensors

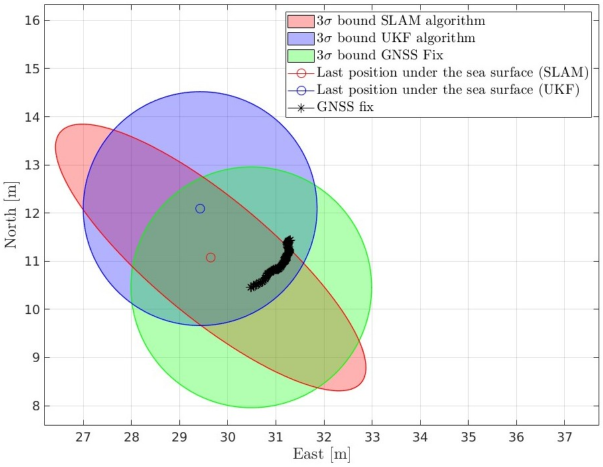


Figure 6.15: 3σ bound of the last positions under the sea surface estimated with the SLAM and UKF algorithms and the first GNSS fix measurement with its accuracy 3σ bound.

between the two filters, the FLS readings have not been employed as speed measurement sources in the UKF. As shown in Fig 6.14, firstly, the AUV navigated over the sea surface to acquire the sensor data for the LG-EKF for the orientation initialization procedure. During this phase, the vehicle followed a straight line from the starting point to the diving point, and when the heading offset covariance estimated through the LG-EKF reached the fixed threshold, the AUV yaw angle was corrected. The goodness of the initialization procedure can be evaluated by comparing the resurfacing error in the presence or absence of the heading correction. Indeed, it is necessary to highlight that the resurfacing error is reduced from 12.07 to 0.899 meters thanks to the initialization procedure.

The position resurfacing error values and covariances have been evaluated on the North-East plane. Fig. 6.14 and Fig. 6.15 respectively report the estimated trajectories and an analysis of the resurfacing errors with their 3σ bound. From Tab. 6.1, analyzing the results from the GNSS resurfacing error, it is easily noticeable that both the proposed strategies are acceptable in terms of navigation estimation quality. To evaluate the agreement between estimation errors and estimated uncertainty, the 3σ bounds during the resurfacing phase are presented. This is summarized in Fig. 6.15, where the 3σ bounds for the filters and the GNSS are presented. In all the analyzed cases, the position provided by the

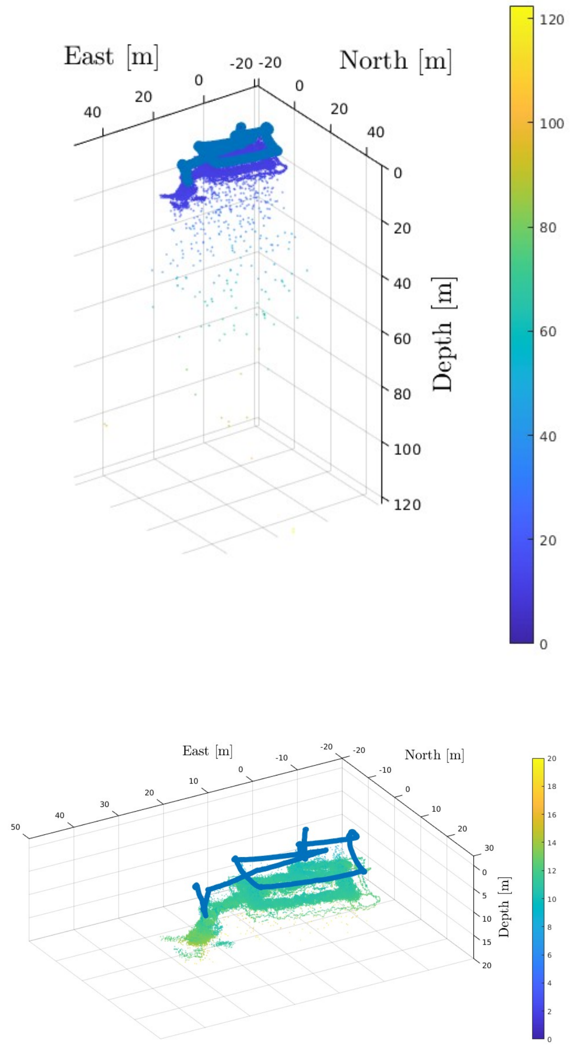


Figure 6.16: Representation of the point cloud and the travelled trajectory estimated through the SLAM algorithm during the mission in Stromboli Island, Messina (Italy). While on the top image the entire point cloud is reported and, due to the presence of outliers, the depth scale is too extended, on the bottom image a zoom on the region of interest is performed.

filter (with its confidence bounds) appears to guarantee a reasonable prediction of the vehicle's true position when it resurfaces. The employed GNSS has an expected accuracy on the order of meters and the 2D error can be represented as a 2D Gaussian distribution whose components are independently distributed.

Table 6.1: Navigation performance for the mission accomplished in Stromboli Island, Messina (Italy): resurfacing error.

Navigation strategy	Error [m]
UKF algorithm	1.943
SLAM algorithm	0.899

Focusing the attention on the SLAM algorithm and its mapping capabilities, Fig. 6.16 reports the SLAM-based estimated trajectory and the generated point cloud. It is possible to evaluate the algorithm mapping capabilities by comparing the estimated point cloud with a bathymetry of the region around the island. As for the test in simulated environments, several outliers are kept in the point cloud during the SLAM algorithm operation, which negatively influences the seabed reconstruction. Consequently, the seabed reconstruction capabilities of the developed algorithm and the comparison with the ground truth bathymetry are reported in Appendix A, where the employed post-processing strategies are described.

Table 6.2: Estimated scale factor and rototranslation transform between DVL and camera reference systems.

Parameter	Initial guess	Value after step 1	Value after step 2
s	0.0	5.448	5.529
ϕ_c^b [deg]	0.0	0.0	-0.005
θ_c^b [deg]	90.0	90.0	89.477
ψ_c^b [deg]	0.0	10.119	8.43
$t_{c,b}^x$ [m]	0.24	0.24	0.233
$t_{c,b}^y$ [m]	0.07	0.06	0.076
$t_{c,b}^z$ [m]	0.05	0.05	0.049

The scale factor computation procedure has been applied to estimate the scale factor between the DVL-based trajectory and the visual part of the algo-

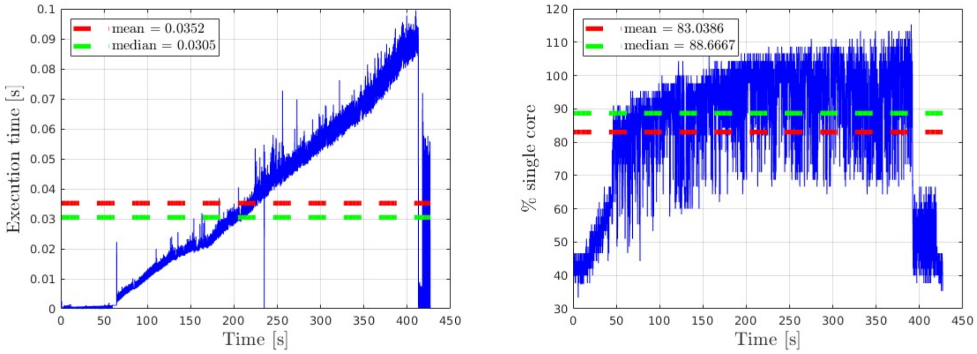


Figure 6.17: On the left, execution time of the SLAM filter, calculated at each iteration as the sum of the requested time for measurement insertion in the factor graph and for the optimization process. On the right, CPU burden analysis. In red and green are respectively reported the mean and the median.

rithm before fusing them in the whole factor graph. In particular, approximate values of the relative position and orientation between the DVL and the camera has been provided as input to the algorithm, but their values have been kept as variables in the optimization process. The scale factor between the DVL-based trajectory and the visual SLAM has been solved with the developed algorithm, and the results have been reported in Tab. 6.2. It is necessary to highlight that the proposed strategy can compensate the alignment error between the camera and the DVL frames. Indeed, due to uncontrollable external conditions (e.g., loosening of the screws during the vehicle preparation, collisions during the diving procedure), the camera rotated around its z-axis during the autonomous mission of an unknown quantity which has been estimated and compensated by the algorithm. The resurfacing error value is equal to 0.899 meters, indicating a high navigation accuracy of the proposed strategy with respect to the GNSS fixes obtained when the vehicle resurfaced.

Finally, regarding the computational burden, the execution time of the filter has been subject of the analysis. The sum of the requested time to perform the measurement insertion in the factor graph and the optimization process has been considered. For what concern the CPU analysis, the output of the command *top* has been recorded to store the data. The results can be found in Fig. 6.17. It is necessary to notice that the instants where the visual part of the algorithm is initialized and stopped can be easily highlighted thanks to its influence on the execution time of each iteration. Indeed, despite the SLAM algorithm optimizes only the last nodes thanks to the properties of the iSAM2 library, handling a continuously growing point cloud increases the required computational cost. When the vehicle resurfaces and the visual part of the algorithm is excluded due to visibility limitations, the necessary computational burden drastically decreases. Indeed, the point cloud is saved, and only the position nodes are updated when new measurements are acquired.

6.5 Main contributions

An underwater visual acoustic SLAM strategy which integrates DVL with a visual SLAM system has been developed to perform accurate navigation and mapping tasks at the same time. Particular attention has been focused on the design of scale factor ambiguity resolution and extrinsic calibration optimization procedure and on implementing a reset procedure to reduce the computational burden. Furthermore, the proposed strategy has been tested with both simulated and experimental data to evaluate the navigation performance and has been compared with the UKF-based algorithm discussed in the previous section.

Chapter 7

Conclusions

This thesis summarizes the results carried out during the Ph.D. period in the years 2019-2022 for what concerns the research activity on underwater navigation and mapping of the MDM Lab of the UNIFI DIEF. The main goal was to improve and enhance the UNIFI DIEF navigation solutions developed during the previous years and to implement novel algorithms to combine navigation and mapping tasks. More in detail, including payload sensors as acoustic and optical sensors in the navigation algorithms and finding the optimal solutions to fuse their measurements, have arisen as the major development guideline.
Starting from the analysis of the state-of-the-art regarding the employment of non-conventional sensors for underwater navigation purposes and fusion approaches in localization algorithms, novel strategies have been developed and compared to evaluate the pros and cons. Indeed, how to suitably fuse the research background of the UNIFI DIEF in underwater navigation with novel requirements in terms of autonomous navigation and mapping has been analyzed.
Firstly, to face against orientation issues during experimental campaigns, an initialization procedure to be performed when the vehicle is on the sea surface to replace the magnetometer measurements and to evaluate the offset caused by the unknown disturbances has been implemented. The proposed solution has been successfully applied to the navigation data acquired during missions performed by FeelHippo AUV in Cecina, Livorno (Italy), where the vehicle was employed for a surveillance mission near the Melania shipwreck, and in La Spezia (Italy), where experimental tests in a harbor have been performed.
Subsequently, a set of novel UKF-based sensor fusion strategies for autonomous

underwater navigation to increase the navigation performance and to guarantee that the AUV can correctly perform its mission and accomplish its tasks has been developed. The centralized strategies guarantee the best improvements in terms of estimation quality, which can be retrieved by analyzing the resurfacing error and the adherence to the reference input trajectory. Decentralized strategies provide an increase of robustness against measurement reduction. In particular, parallel local filtering approaches are less affected by the lack of measurements than centralized algorithms. A preliminary research and testing phase were performed, where simulated data have been employed for testing and validation. Then, an experimental activity at sea, were two underwater missions at Vulcano Island, Messina (Italy) have been accomplished, allowed verifying the goodness of the proposed strategy during real underwater missions.

Kalman filtering condenses the vehicle's history in the last estimate and covariance. A MAP strategy based on factor graphs has been developed to overcome these limitations and include visual landmarks in the estimation process. Visual features are sometimes difficult to be found in underwater environments due to visibility and texture issues. Consequently, the strategy fuses DVL measurements with a visual SLAM system to simultaneously perform accurate navigation and mapping tasks. DVL beam data can be employed for speed measurement and to obtain an approximated knowledge of the sea bottom. As for the UKF-based filters, both simulated and experimental data have been employed to evaluate the capabilities of the developed strategy. The experimental data have been acquired during trials at Stromboli Island, Messina (Italy), using the vehicle FeelHippo AUV.

In conclusion, this thesis presented the development of strategies to improve the navigation algorithm of FeelHippo AUV, by exploiting also payload sensors available onboard, with the aim of increasing the localization and mapping capabilities of the vehicle.

During the experimental campaign, FeelHippo AUV was the only vehicle involved; nevertheless, since the proposed solutions are not tailored to a particular vehicle, their outcomes can be deemed as general, and future developments will include the testing of the navigation strategy on other vehicles. Furthermore, progresses on the developed algorithms still needs to be made. Integrating the developed estimation strategies within the attitude estimator could represent an important subject to be investigated. Indeed, developing a unique filter that works on both the attitude and position estimation in a coupled way could increase navigation precision. Concerning the SLAM-based algorithm, including acoustic FLS images in the pose graph framework would push forward the performance of the navigation filter. Despite the intrinsic characteristic of FLS (low resolution, influence of the viewpoint) images poses relevant issues to face, the employment of an additional acoustic sensor can be useful to apply the developed strategy in scenarios with reduced visibility. Finally, from a mapping-based point of view, a multi-vehicle solution for autonomously fusing the underwater environment reconstructions could represent a coherent continuation of the research activity carried out so far. The proposed SLAM strategy could operate onboard of each vehicle and, by employing relative or absolute position measurements, the estimated maps could be fused in a unique more detailed reconstruction.

Appendix A

Seabed 3D reconstruction and texture application

Mapping the surrounding environment is a common task in underwater exploration, and it is fundamental to enhance the vehicle capabilities to find objects of potential interest. The point clouds obtained from the SLAM algorithm have been processed with an automatic tool to obtain a 3D reconstruction of the sea bottom. The developed reconstruction strategy takes as input the estimated point cloud and the geographical coordinate of a reference point and automatically generates a 3D reconstruction with a simple texturization procedure and a georeferenced depth map, thanks to the employment of the functions implemented in the open-source libraries Point Cloud Library (PCL) [91] and Open3d [92].

Analyzing the point cloud obtained from the navigation algorithm applied in both simulated and real environments, it is necessary to notice that some points can be classified as outliers. Therefore the need arises to eliminate them as the displayed graphs are excessively bulky and negatively influence the mesh realization. For each point, a fixed number of neighbors is defined to estimate the mean of the average distance, and a point is considered an outlier if the average distance to its neighbors is above a specified threshold [93]. The outlier eliminating process, therefore, leads to a significant decrease in points, making the representations more uniform. Subsequently, the point cloud is processed with a smoothing method to filter out the noise of the measurements on the processed points. In particular, median filtering of the 3D point cloud data is

performed.
The 3D mesh generation algorithm is applied to the filtered point cloud. The Poisson algorithm [94] is applied, and its parameters have been set to optimize the reconstruction process. It is necessary to note that the depth value and the limit density of the points at which the reconstruction is cut have been chosen to compromise reconstruction speed and estimation quality. The advantages of the Poisson algorithm over other surface fitting methods are numerous. Many implicit methods of surface fitting segment the data into regions for local fitting and further combine these local approximations using blending functions. In contrast, Poisson reconstruction is a global solution that considers all the data simultaneously without resorting to heuristic partitioning or blending. In this way, Poisson reconstruction creates very smooth surfaces that robustly approximate noisy data.

Table A.1: Mean with the associated covariance and median values of the depth error in presence and in absence of the filtering procedure.

Parameter	Before filtering	After filtering
Mean [m]	0.2767	0.2002
Covariance [m]	7.6212	0.0386
Median [m]	0.1469	0.1465

Finally, the texture is applied to the mesh. Providing as inputs the mesh and the camera positions and directions obtained from the SLAM algorithm, the acquired images are applied to the mesh thanks to the projection model of the camera. To apply this algorithm, it is necessary to know the camera projection model (as the focal length and the principal point coordinate, which can be retrieved from the camera calibration procedure) and the image size.
Firstly, considering that in the simulated environment created with UUV Simulator the seabed can be generated with a mathematical function $z = f(x, y)$, it is possible to evaluate the performance of the filtering algorithm. It is also necessary to notice that the simulated seabed has been textured with an image rich in features to facilitate the correct behavior of the visual part of the SLAM algorithm. Fig. A.1 reports the 3D filtered point cloud with the estimated trajectory. It is necessary to compare this point cloud with the one directly obtained from the SLAM algorithm and reported in Fig. 6.13. Two error maps have been created with the point clouds, as before and after the filtering procedure, to analyze the improvements in seabed reconstruction. It is necessary to

notice that the outlier points are correctly removed, and the point cloud size is reduced to increase its easiness of management by the reconstruction algorithm (see Fig. A.2 and Fig. A.3). As can be retrieved from Tab. A.1, while the outlier removal process does not influence the mean and the median values due to the high number of points, the covariance associated with the mean value is strongly reduced.

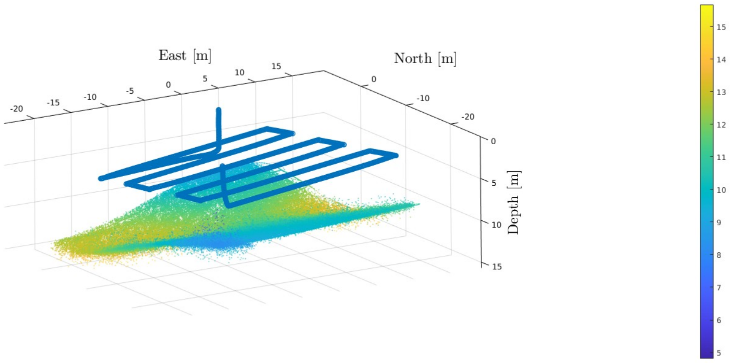


Figure A.1: Filtered point cloud and estimated trajectory travelled by the simulated vehicle.

Finally, the 3D point cloud has been processed with the Poisson reconstruction algorithm to build the 3D mesh. Thanks to the chosen reconstruction algorithm, the obtained mesh is smoothed and correctly follows the shape of the simulated sea bottom (see Fig. A.4). Furthermore, as shown in Fig. A.5, it is possible to apply the texture on the reconstructed surface by saving images and camera positions at a fixed sampling time.

Some of the results obtained during the mission performed in Stromboli Island, Messina (Italy), in September 2022, are presented. In particular, a 3D reconstruction and a geo-localized map of the sea bottom are reported. The reconstruction comprises around $240k$ points obtained as output from the SLAM algorithm. Firstly, the outlier points have been removed (see Fig. A.7), and then, the 3D point cloud has been processed with the Poisson reconstruction algorithm to build the 3D mesh, which is shown in Fig. A.8. Looking at Fig. A.11, the 3D reconstructed mesh has been converted into a depth map and referenced to the island in Google Earth. During this step, the holes in the 3D reconstruction have been filled by employing a natural neighbor interpolation

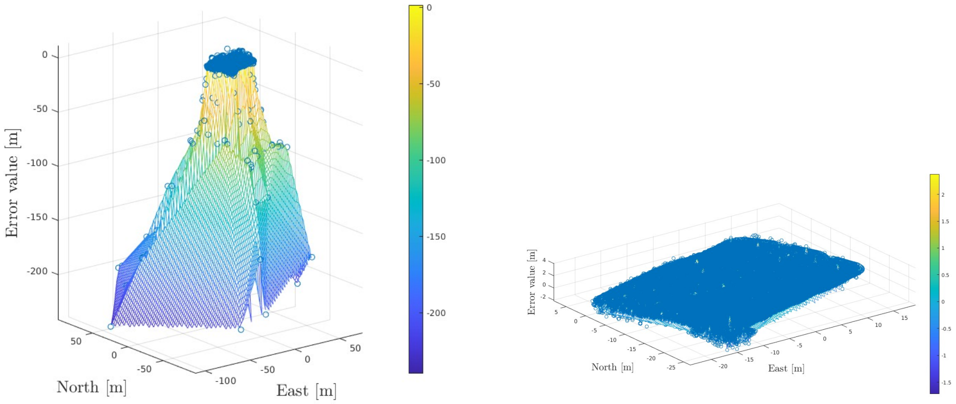


Figure A.2: Representation of the error point clouds computed by comparing the reference sea bed function and the estimated point cloud and generation of the estimated error maps before (top image) and after (bottom image) the filtering procedure.

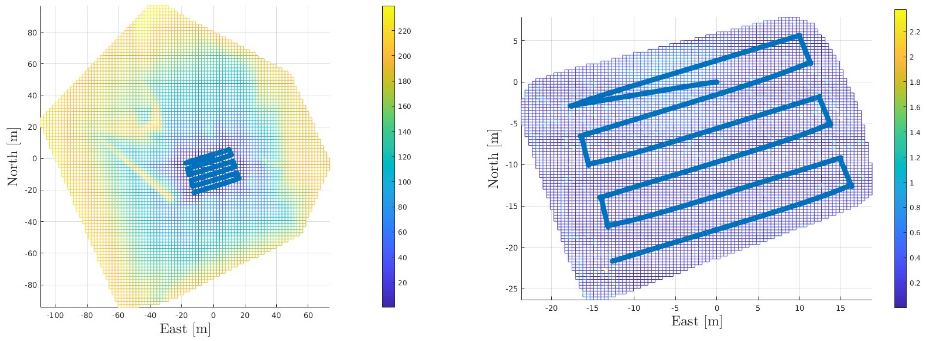


Figure A.3: Comparison between the estimated error maps before (top image) and after (bottom image) the filtering procedure with respect to the travelled trajectory by the simulated vehicle.

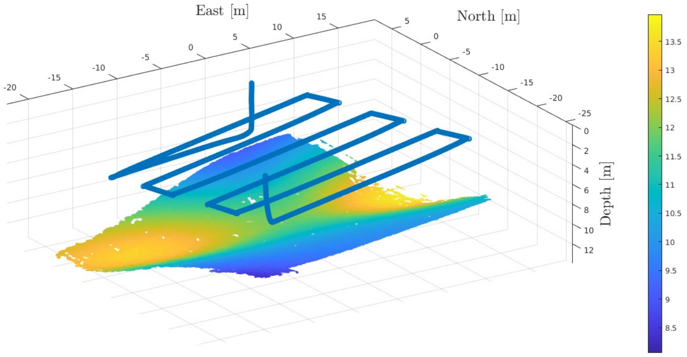


Figure A.4: Resulting sea bottom 3D mesh reconstruction and estimated trajectory travelled by the simulated vehicle.

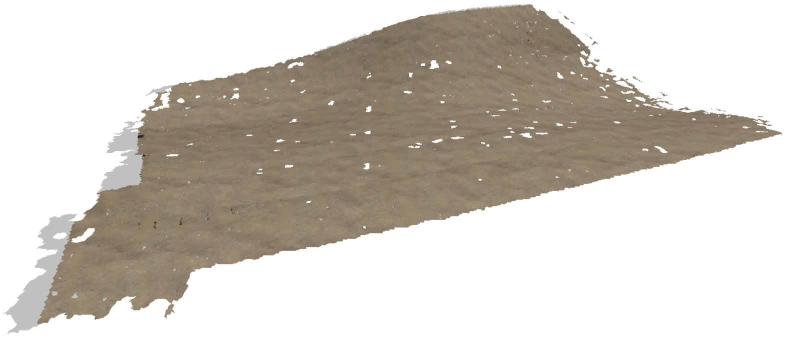


Figure A.5: Resulting sea bottom 3D mesh reconstruction with texture application.

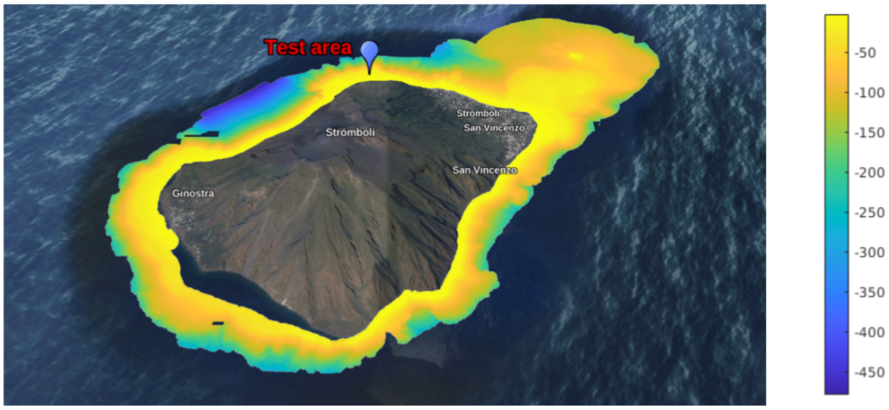


Figure A.6: Reference bathymetry of the sea bottom around Stromboli Island, Messina (Italy). The test area, where FeelHippo AUV performed its autonomous mission, is highlighted.

[95]. In addition, the good matching between the reference bathymetry, whose data have mainly funded by the National Research Council and Presidenza del Consiglio dei Ministri–Dipartimento della Protezione Civile, through specific agreement (see Fig. A.6), and the estimated 3D reconstruction can also be observed to prove the reconstruction's goodness. The provided bathymetry has a horizontal resolution of 5 meters. Thus only an approximate comparison can be performed, but it can be sufficient to have a simple evaluation of the generated point cloud. All the points of the cloud that lies in each square generated from the ground truth bathymetry are employed to compute the mean point and perform the comparison (Fig. A.10).
Finally, as shown in Fig. A.9, the texture has been applied to the reconstructed surface by saving images and camera positions at a fixed sampling time.

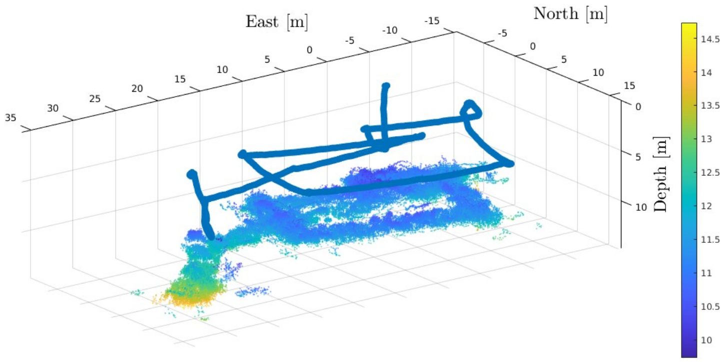


Figure A.7: Filtered point cloud and estimated trajectory travelled by the vehicle during the autonomous mission accomplished in Stromboli Island, Messina (Italy).

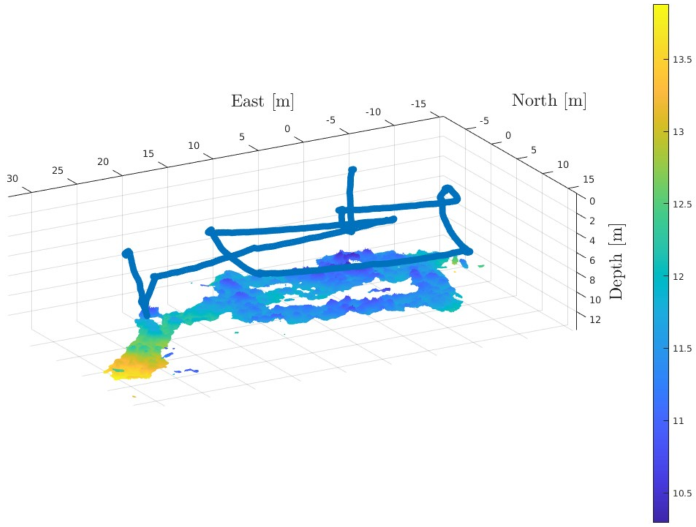


Figure A.8: Resulting sea bottom 3D mesh reconstruction and estimated trajectory travelled by the vehicle during the autonomous mission accomplished in Stromboli Island, Messina (Italy).

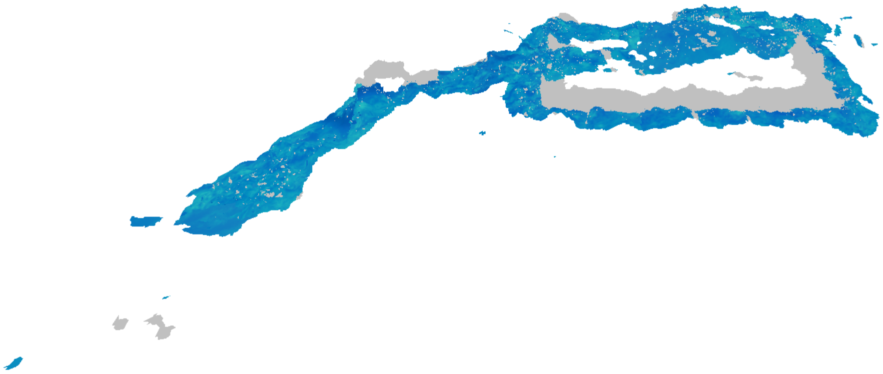


Figure A.9: Resulting sea bottom 3D mesh reconstruction with texture application of the area near Punta Labronzo in Stromboli Island, Messina (Italy).

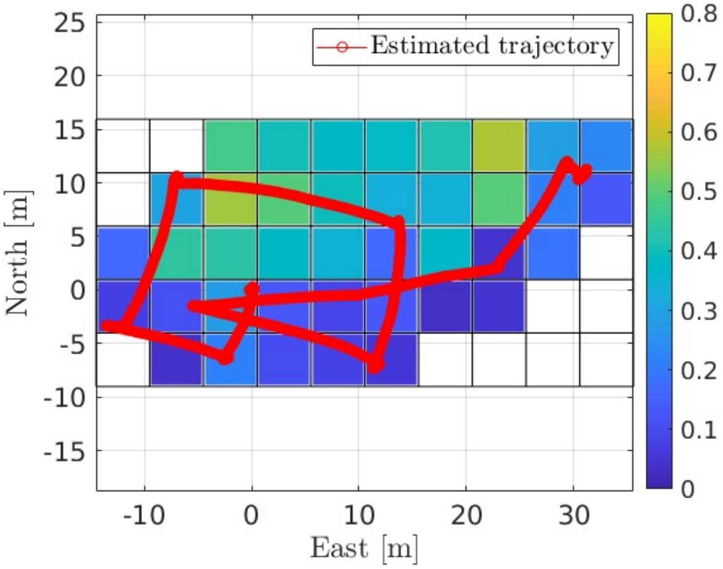


Figure A.10: Estimated error bathymetry map with respect to the trajectory travelled by the vehicle during the autonomous mission accomplished in Stromboli Island, Messina (Italy).

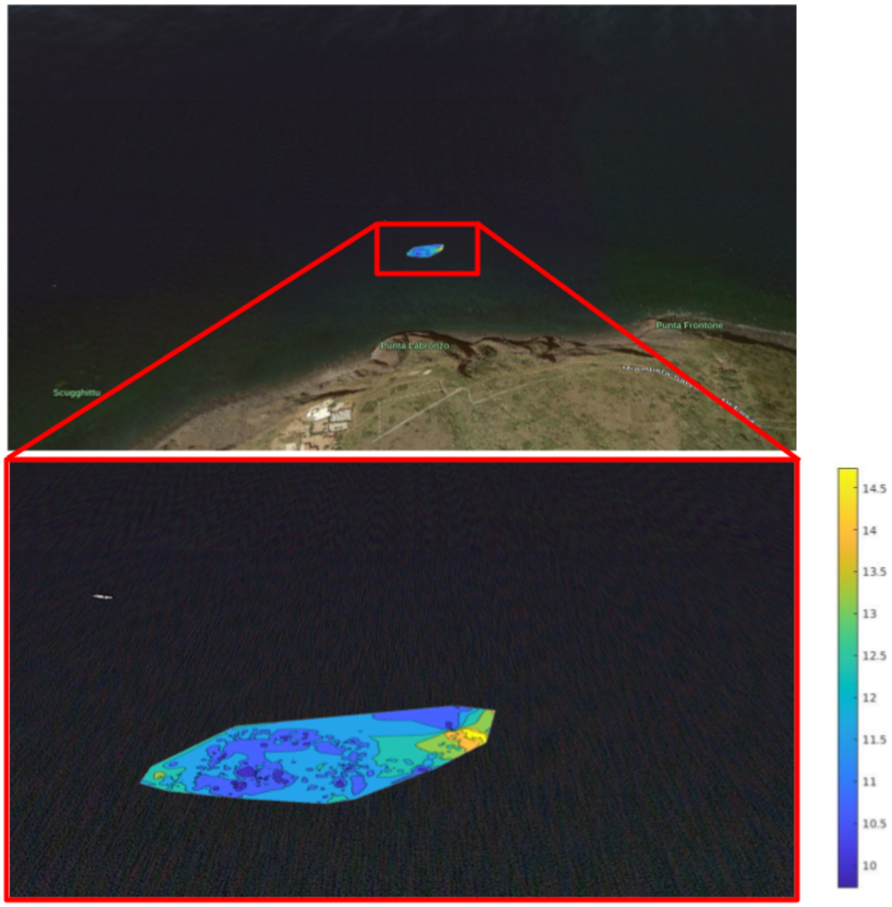


Figure A.11: Google Earth image of the Punta Labronzo area in Stromboli Island, Messina (Italy) overlaid with the underwater depth map generated from the estimated 3D mesh. On the bottom image, a close view of the depth map is presented.

References

[1] M. Prats, D. Ribas, N. Palomeras, J. C. García, V. Nannen, S. Wirth, J. J. Fernández, J. P. Beltrán, R. Campos, P. Ridao, P. J. Sanz, G. Oliver, M. Carreras, N. Gracias, R. Marín, and A. Ortiz, "Reconfigurable AUV for intervention missions: a case study on underwater object recovery," *Intelligent Service Robotics*, vol. 5, pp. 19–31, 2012.

[2] G. Ferri, F. Ferreira, and V. Djapic, "Multi-domain robotics competitions: The CMRE experience from SAUC-E to the European Robotics League Emergency Robots," in *OCEANS 2017 - Aberdeen*, pp. 1–7, 2017.

[3] M. Cashmore, M. Fox, T. Larkworthy, D. Long, and D. Magazzeni, "AUV mission control via temporal planning," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6535–6541, 2014.

[4] D. Youakim, P. Cieslak, A. Dornbush, A. Palomer, P. Ridao, and M. Likhachev, "Multirepresentation, Multiheuristic A* search-based motion planning for a free-floating underwater vehicle-manipulator system in unknown environment," *Journal of Field Robotics*, vol. 37, no. 6, pp. 925–950, 2020.

[5] E. Vidal, N. Palomeras, K. Istenič, N. Gracias, and M. Carreras, "Multisensor online 3D view planning for autonomous underwater exploration," *Journal of Field Robotics*, vol. 37, no. 6, pp. 1123–1147, 2020.

[6] L. Paull, S. Saeedi, M. Seto, and H. Li, "Sensor driven online coverage planning for autonomous underwater vehicles," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2875–2880, 2012.

[7] B. Allotta, R. Conti, R. Costanzi, F. Fanelli, J. Gelli, E. Meli, N. Monni, A. Ridolfi, and A. Rindi, "A low cost autonomous underwater vehicle for

patrolling and monitoring," *Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 231, no. 3, pp. 740–749, 2017.

[8] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[9] L. Zacchini, A. Ridolfi, A. Topini, N. Secciani, A. Bucci, E. Topini, and B. Allotta, "Deep Learning for on-board AUV Automatic Target Recognition for Optical and Acoustic imagery," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14589–14594, 2020.

[10] P. A. Miller, J. A. Farrell, Y. Zhao, and V. Djapic, "Autonomous underwater vehicle navigation," *IEEE Journal of Oceanic Engineering*, vol. 35, no. 3, pp. 663–678, 2010.

[11] M. Bresciani, S. Tani, F. Ruscio, L. Zacchini, L. Bartalucci, A. Ridolfi, F. Maurelli, and R. Costanzi, "Impact of natural gas seeps on the navigation of an autonomous underwater vehicle," in *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, pp. 1–6, 2022.

[12] M. Hildebrandt and F. Kirchner, "IMU-aided stereo visual odometry for ground-tracking AUV applications," in *OCEANS'10 IEEE SYDNEY*, pp. 1–8, 2010.

[13] F. Shkurti, I. Rekleitis, M. Scaccia, and G. Dudek, "State estimation of an underwater robot using visual and inertial information," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5054–5060, 2011.

[14] Y. Yang and G. Huang, "Acoustic-inertial underwater navigation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4927–4933, 2017.

[15] M. Franchi, A. Ridolfi, and M. Pagliai, "A forward-looking sonar and dynamic model-based AUV navigation strategy: Preliminary validation with FeelHippo AUV," *Ocean Engineering*, vol. 196, p. 106770, 2020.

[16] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[17] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley and Sons, 2004.

[18] H. Hong, J. G. Lee, C. G. Park, and H. S. Han, “A leveling algorithm for an underwater vehicle using extended Kalman filter,” in *IEEE 1998 Position Location and Navigation Symposium*, pp. 280–285, 1998.

[19] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[20] B. Allotta, A. Caiti, R. Costanzi, F. Fanelli, D. Fenucci, E. Meli, and A. Ridolfi, “A new AUV navigation system exploiting Unscented Kalman Filter,” *Ocean Engineering*, vol. 113, pp. 121–132, 2016.

[21] R. Costanzi, F. Fanelli, E. Meli, A. Ridolfi, A. Caiti, and B. Allotta, “UKF-based navigation system for AUVs: Online experimental validation,” *IEEE Journal of Oceanic Engineering*, vol. 44, no. 3, pp. 633–641, 2018.

[22] S. Song, B. Kim, J. Kim, and S.-C. Yu, “Real-Time Reliability Evaluation of Optical and Acoustic Images for Feature-Based Localization of AUV,” in *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*, pp. 1–6, IEEE, 2018.

[23] P. Y. C. H. Robert Grover Brown, *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley Sons, 2012.

[24] D. Simon, “Alternate Kalman filter formulations,” in *Optimal State Estimation*, ch. 6, pp. 149–181, John Wiley Sons, Ltd, 2006.

[25] N. A. Carlson, “Federated square root filter for decentralized parallel processors,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 3, pp. 517–525, 1990.

[26] N. A. Carlson and M. P. Berarducci, “Federated Kalman filter simulation results,” *Navigation, Journal of the institute of navigation*, vol. 41, no. 3, pp. 297–322, 1994.

[27] Y. Li and Y. Q. Jiang, “The Application of Distributed Multi-Sensor Information Fusion Technology in Underwater Vehicle,” in *Materials Science and Information Technology II*, vol. 532, pp. 1006–1010, 2012.

[28] W. Lyu, X. Cheng, and J. Wang, “Adaptive Federated IMM Filter for AUV Integrated Navigation Systems,” *Sensors*, vol. 20, no. 23, p. 6806, 2020.

[29] S. Grime and H. F. Durrant-Whyte, “Data fusion in decentralized sensor networks,” *Control engineering practice*, vol. 2, no. 5, pp. 849–863, 1994.

[30] R. Lobbia and M. Kent, “Data fusion of decentralized local tracker outputs,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 3, pp. 787–799, 1994.

[31] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in *2007 46th IEEE Conference on Decision and Control*, pp. 5492–5498, 2007.

[32] G. Battistelli and L. Chisci, “Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability,” *Automatica*, vol. 50, no. 3, pp. 707 – 718, 2014.

[33] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, “Discrete-time distributed Kalman filter design for multi-vehicle systems,” in *2017 American Control Conference (ACC)*, pp. 5538–5543, 2017.

[34] D. Viegas, P. Batista, P. Oliveira, and C. Silvestre, “Discrete-time distributed Kalman filter design for formations of autonomous vehicles,” *Control Engineering Practice*, vol. 75, pp. 55–68, 2018.

[35] F. Dellaert, “Factor graphs: Exploiting structure in robotics,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 141–166, 2021.

[36] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[37] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-Manifold Preintegration for Real-Time Visual–Inertial Odometry,” *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.

[38] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental smoothing and mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[39] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[40] G. Grisetti, T. Guadagnino, I. Aloise, M. Colosi, B. Della Corte, and D. Schlegel, "Least squares optimization: From theory to practice," *Robotics*, vol. 9, no. 3, p. 51, 2020.

[41] E. Westman and M. Kaess, "Degeneracy-aware imaging sonar simultaneous localization and mapping," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 4, pp. 1280–1294, 2019.

[42] M. Franchi, A. Bucci, L. Zacchini, A. Ridolfi, M. Bresciani, G. Peralta, and R. Costanzi, "Maximum A Posteriori estimation for AUV localization with USBL measurements," *IFAC-PapersOnLine*, vol. 54, no. 16, pp. 307–313, 2021. 13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2021.

[43] M. F. Fallon, J. Folkesson, H. McClelland, and J. J. Leonard, "Relocating Underwater Features Autonomously Using Sonar-Based SLAM," *IEEE Journal of Oceanic Engineering*, vol. 38, no. 3, pp. 500–513, 2013.

[44] T. A. Huang and M. Kaess, "Towards acoustic structure from motion for imaging sonar," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 758–765, IEEE, 2015.

[45] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[46] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[47] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[48] S. Hong and J. Kim, "Three-dimensional Visual Mapping of Underwater Ship Hull Surface Using Piecewise-planar SLAM," *International Journal of Control, Automation and Systems*, vol. 18, pp. 564–574, 2020.

[49] P. Du, J. Han, J. Wang, G. Wang, D. Jing, X. Wang, and F. Qu, “View-based underwater SLAM using a stereo camera,” in *OCEANS 2017 - Aberdeen*, pp. 1–6, 2017.

[50] S. Rahman, A. Q. Li, and I. Rekleitis, “Sonar Visual Inertial SLAM of Underwater Structures,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5190–5196, 2018.

[51] S. Rahman, A. Q. Li, and I. M. Rekleitis, “SVIn2: Sonar Visual-Inertial SLAM with Loop Closure for Underwater Navigation,” *ArXiv*, vol. abs/1810.03200, 2018.

[52] P. Ozog and R. M. Eustice, “Real-time SLAM with piecewise-planar surface models and sparse 3D point clouds,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1042–1049, 2013.

[53] A. Kim and R. M. Eustice, “Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.

[54] E. Westman and M. Kaess, “Underwater AprilTag SLAM and calibration for high precision robot localization,” Tech. Rep. CMU-RI-TR-18-43, Carnegie Mellon University, Pittsburgh, PA, 2018.

[55] J. Solà, J. Deray, and D. Atchuthan, “A micro Lie theory for state estimation in robotics,” 2018.

[56] I. Marković, J. Ćesić, and I. Petrović, “On wrapping the Kalman filter and estimating with the SO(2) group,” in *2016 19th International Conference on Information Fusion (FUSION)*, pp. 2245–2250, 2016.

[57] G. de Moura Magalhães, E. Dranka, Y. Cáceres, J. B. R. do Val, and R. S. Mendes, “EKF on Lie groups for radar tracking using polar and Doppler measurements,” in *2018 IEEE Radar Conference (RadarConf18)*, pp. 1573–1578, 2018.

[58] T. Fossen, *Guidance and Control of Ocean Vehicles.* London, UK, Wiley, 1994.

[59] J. Newman, *Marine hydrodynamics.* MIT press, 2018.

[60] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, vol. 3068, pp. 182 – 193, 1997.

[61] L. W. Barclay, *Propagation of radiowaves.* Institution of Electrical Engineers, 2003.

[62] F. M. Mirzaei and S. I. Roumeliotis, "A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1143–1156, 2008.

[63] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning.* The MIT Press, 2009.

[64] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[65] E. Westman and M. Kaess, "Degeneracy-Aware Imaging Sonar Simultaneous Localization and Mapping," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 4, pp. 1280–1294, 2020.

[66] R. Costanzi, F. Fanelli, N. Monni, A. Ridolfi, and B. Allotta, "An Attitude Estimation Algorithm for Mobile Robots Under Unknown Magnetic Disturbances," *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 1900–1911, 2016.

[67] B. Allotta, R. Costanzi, F. Fanelli, N. Monni, and A. Ridolfi, "Underwater Vehicles attitude estimation in presence of magnetic disturbances," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2612–2617, 2016.

[68] B. Allotta, A. Caiti, L. Chisci, R. Costanzi, F. Di Corato, C. Fantacci, D. Fenucci, E. Meli, and A. Ridolfi, "An unscented Kalman filter based navigation algorithm for autonomous underwater vehicles," *Mechatronics*, vol. 39, pp. 185–195, 2016.

[69] A. Bucci, L. Zacchini, and A. Ridolfi, "EKF on Lie Groups for Autonomous Underwater Vehicles orientation initialization in presence of magnetic disturbances," in *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV 2022)*, pp. 1–6, 2022.

[70] B. Fan, Q. Li, and T. Liu, "How Magnetic Disturbance Influences the Attitude and Heading in Magnetic and Inertial Sensor-Based Orientation Estimation," *Sensors*, vol. 18, no. 1, 2018.

[71] Y. S. Suh, Y. S. Ro, and H. J. Kang, "Quaternion-Based Indirect Kalman Filter Discarding Pitch and Roll Information Contained in Magnetic Sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1786–1792, 2012.

[72] A. Bucci, M. Franchi, A. Ridolfi, N. Secciani, and B. Allotta, "Evaluation of UKF-Based Fusion Strategies for Autonomous Underwater Vehicles Multisensor Navigation," *IEEE Journal of Oceanic Engineering*, vol. 48, no. 1, pp. 1–26, 2023.

[73] F. Bellavia, M. Fanfani, F. Pazzaglia, C. Colombo, R. Costanzi, N. Monni, A. Ridolfi, and B. Allotta, "Piecewise planar underwater mosaicing," in *OCEANS 2015 - Genova*, pp. 1–7, 2015.

[74] X. Li, M. Shang, H. Qin, and L. Chen, "Fast accurate fish detection and recognition of underwater images with Fast R-CNN," in *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–5, 2015.

[75] C. Beall, B. J. Lawrence, V. Ila, and F. Dellaert, "3D reconstruction of underwater structures," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4418–4423, 2010.

[76] L. Zacchini, A. Bucci, M. Franchi, R. Costanzi, and A. Ridolfi, "Mono visual odometry for Autonomous Underwater Vehicles navigation," in *2019 MTS/IEEE Oceans, Marseille, France*, 2019.

[77] A. Bucci, L. Zacchini, M. Franchi, A. Ridolfi, and B. Allotta, "Comparison of feature detection and outlier removal strategies in a mono visual odometry algorithm for underwater navigation," *Applied Ocean Research*, vol. 118, p. 102961, 2022.

[78] N. Hurtós, D. Ribas, X. Cufí, Y. Petillot, and J. Salvi, "Fourier-based Registration for Robust Forward-looking Sonar Mosaicing in Low-visibility Underwater Environments," *Journal of Field Robotics*, vol. 32, no. 1, pp. 123–151, 2015.

[79] M. Franchi, A. Ridolfi, and L. Zacchini, "A Forward-Looking Sonar-Based for Underwater Mosaicing and Acoustic Odometry," in *2018 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV 2018)*, pp. 1–6, 2018.

[80] T. Guerneve, K. Subr, and Y. Petillot, “Three-dimensional reconstruction of underwater objects using wide-aperture imaging SONAR,” *Journal of Field Robotics*, vol. 35, no. 6, pp. 890–905, 2018.

[81] M. Valdenegro-Toro, “Deep Neural Networks for Marine Debris Detection in Sonar Images,” 2019. arXiv preprint arXiv:1905.05241.

[82] M. D. Aykin and S. Negahdaripour, “Forward-look 2-D sonar image formation and 3-D reconstruction,” in *2013 OCEANS - San Diego*, pp. 1–10, 2013.

[83] M. Franchi, A. Ridolfi, L. Zacchini, and B. Allotta, “Experimental evaluation of a Forward-Looking SONAR-based system for acoustic odometry,” in *2019 MTS/IEEE Oceans, Marseille, France*, 2019.

[84] D. Willner, C. B. Chang, and K. P. Dunn, “Kalman filter algorithms for a multi-sensor system,” in *1976 IEEE Conference on Decision and Control including the 15th Symposium on Adaptive Processes*, pp. 570–574, 1976.

[85] J. Kim, G. I. Jee, and J. G. Lee, “A Federated Kalman Filter design using a gain fusion algorithm,” *IFAC Proceedings Volumes*, vol. 31, pp. 385–391, 1998.

[86] J. Salvi, Y. Petillot, and E. Batlle, “Visual SLAM for 3D large-scale seabed acquisition employing underwater vehicles,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1011–1016, 2008.

[87] A. Mallios, P. Ridao, D. Ribas, F. Maurelli, and Y. Petillot, “EKF-SLAM for AUV navigation under probabilistic sonar scan-matching,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4404–4411, 2010.

[88] F. Dellaert, “Factor graphs and GTSAM: A hands-on introduction,” tech. rep., Georgia Institute of Technology, 2012.

[89] S. Umeyama, “Least-squares estimation of transformation parameters between two point patterns,” *IEEE Computer Architecture Letters*, vol. 13, no. 04, pp. 376–380, 1991.

[90] A. Björck, *Numerical Methods for Least Squares Problems.* Society for Industrial and Applied Mathematics, 1996.

[91] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 9-13 2011.

[92] Q. Y. Zhou, J. Park, and V. Koltun, “Open3D: A modern library for 3D data processing,” *arXiv:1801.09847*, 2018.

[93] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, “Towards 3D Point cloud based object maps for household environments,” *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 927–941, 2008.

[94] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson Surface Reconstruction,” in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, p. 61–70, 2006.

[95] R. Sibson, “A Brief Description of Natural Neighbor Interpolation,” in *Interpreting Multivariate Data* (V. Barnett, ed.), pp. 21–36, New York: John Wiley Sons, 1981.