

# Underwater Geophysical Navigation using a Particle Filter Approach to Multi-Sensor Fusion

Marcelo Jacinto\*, André Potes\*, David Souto\*, Yusen Gong<sup>†</sup>,

João Quintas\*, João Cruz\*, Shubham Garg\* and António Pascoal\*

\*Laboratory of Robotics and Systems in Engineering and Science (LARSyS),  
ISR/IST, University of Lisbon, Lisbon, Portugal.

Email: marcelo.jacinto@tecnico.ulisboa.pt

<sup>†</sup>Science and Technology on Underwater Vehicles Laboratory, Harbin Engineering University, Harbin, China

**Abstract**—This paper addresses the problem of underwater navigation of autonomous underwater vehicles via a Monte Carlo estimation approach that relies on the use of a prior digital elevation map of the seabed and bathymetric data acquired with a Multibeam Echosounder. The Monte Carlo estimation procedure is implemented in the form of a particle filter, as part of a multi-sensor fusion framework for underwater geophysical navigation. The contribution of this paper focuses on two main topics: i) development of a particle filter as a solution to a terrain-based Bayesian vehicle positioning problem, followed by filter implementation and simulation and ii) integration of the particle filter structure in a multi-sensor fusion architecture for vehicle navigation with a view to increasing navigation accuracy. The results of realistic simulations with a dedicated marine robotics system simulator illustrate the navigation performance achieved with the proposed solution.

**Index Terms**—Monte Carlo positioning, Particle Filter, Kalman Filter, Sensor Fusion, Bathymetric Geophysical Navigation, Multibeam Echosounder, DVL and AHRS.

## I. INTRODUCTION

Recent years have witnessed tremendous growth in ocean exploration activities. The increasing knowledge about the marine environment, largely inaccessible to humans, has been occurring in parallel with technological advances in Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs), making them the tools par excellence for the execution of scientific and commercial missions at sea. The continuous improvement of these vehicles has always been accompanied by new challenges that arise in the design and development of new control and navigation systems with a view to increasing their autonomy and reliability.

Underwater positioning concerns a robot's capability to know its whereabouts in the surrounding environment. This is a challenging and important problem in the current ocean exploration scene, which has witnessed growing interest due to the demand for increasingly more accurate navigation systems for ROVs and AUVs, especially when operating at large ocean depths.

This work was partially supported by the H2020 EU Marine Robotics Research Infrastructure Network (Project ID 731103), the H2020-EU.1.2.2 FET Proactive RAMONES project (Grant agreement ID: 101017808), and the ANI/PT 2020 Ocean-Tech project. Funded was also provided by FCT, FEDER, and ERDF through the LARSyS UIDB/50009/2020, LISBOA-01-0145-FEDER-031411, and EMSO-PT projects.

For navigation purposes, the vast majority of AUVs rely heavily on the integration of dead reckoning (DR) data obtained via sensors such as Doppler Velocity Loggers (DVL) and Attitude and Heading Reference Systems (AHRS). This method, on its own, is prone to errors due to presence of ocean currents and sensors bias and drift that are integrated over time, as the vehicle executes a mission [1]. Furthermore, access to a Global Positioning System (GPS) for submerged vehicles is impracticable, due to the fact that electromagnetic waves are heavily attenuated in the water medium, and position fixes utilizing this system require the vehicle to resurface periodically. Alternative strategies rely on getting position fixes obtained with an Ultra-Short Baseline (USBL) or a Long Baseline (LBL) system to improve the accuracy of navigation. These systems utilize acoustic waves to position a vehicle [2], [3] or, in some applications, to localize an underwater target [4]. In spite of their capability to substantially improve the performance that can be achieved with DR only, the above systems involve the use of equipment that is costly and requires painstaking deployment and calibration procedures. Furthermore, such system are still prone to latency issues, multipath effects, and low interrogation rates. Another common approach used to bound the accumulated errors is to resort to geophysical navigation methods which rely on matching local measurements of geophysical data such as bathymetric, gravimetric or geomagnetic with a prior geophysical map of the area of interest [5]. In [6] the authors carried out a practical comparison of different filtering techniques applied to geophysical navigation using geomagnetic data. Common to all these strategies is the presence of noise in the sensors measurements, from which emerges the need for a robust method to filter and combine all the information available to the vehicle for navigation and control purposes.

The current work addresses two key technical problems: underwater vehicle positioning and navigation. A Monte Carlo estimation method, referred to as a Particle Filter (PF) estimator, is proposed as a solution to the positioning problem. This is done by resorting to an a priori digital elevation map (DEM) of the seafloor and a Multibeam Echosounder providing measurements of slant ranges to the

terrain, complemented with data obtained from an on board suite of sensors, to yield estimates of the vehicle's position in an inertial reference frame. Lastly, as a solution to the more general navigation problem, the PF position estimates are incorporated into a broader sensor fusion framework used for underwater navigation [7], [8]. This not only provides the particle filter with the necessary odometry data filtered by the navigation system, allowing for a more accurate estimate of the vehicle position, but also creates an additional source of positioning to the vehicle navigation system.

The proposed solution is evaluated with the help of the UUV simulator [9], an extension of the very popular Gazebo simulator [10], using an in-built dynamic model of the Rexrov built [11] and a model of the MEDUSA class of vehicles [12], property of ISR-DSOR.

## II. PROBLEM DESCRIPTION AND STOCHASTIC MODEL

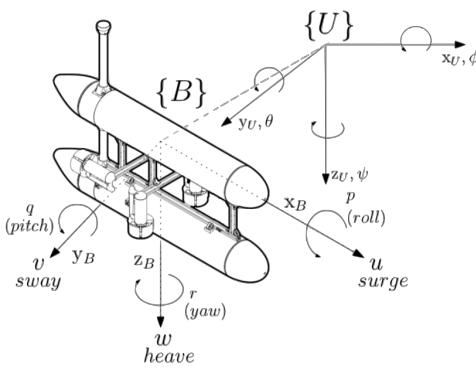
Motivated by the results described in [13], the problem of terrain-based vehicle positioning addressed in this work is that of estimating the position of a vehicle using a particle filter solution, assuming that the vehicle moves in a known 2D horizontal plane at constant depth.

### A. Problem Framework

Let  $\{U\}$  denote an inertial reference frame and  $\{B\}$  a body-fixed reference frame rigidly attached to the geometric center of mass of the vehicle, according to Fig. 1. Let  $\hat{\mathbf{x}} = [\hat{x}, \hat{y}]^T \in \mathbb{R}^2$  denote the position estimate of the origin of  $\{B\}$  in  $\{U\}$ . The kinematic motion of the vehicle, in a 2D horizontal plane, is described by

$$\begin{cases} \dot{\hat{x}} = u \cdot \cos(\psi) - v \cdot \sin(\psi) \\ \dot{\hat{y}} = u \cdot \sin(\psi) + v \cdot \cos(\psi) \end{cases}, \quad (1)$$

where  $u \in \mathbb{R}$  and  $v \in \mathbb{R}$  are the surge and sway linear speeds, respectively of  $\{B\}$  relative to the water and  $\psi \in \mathbb{R}$  denotes the heading angle of  $\{B\}$  expressed in  $\{U\}$ . The surge and sway linear velocities are assumed to be available via a DVL and the heading angle is obtained through an AHRS. These three variable serve as inputs to the positioning filter. Unmeasured ocean currents are not taken into consideration in this motion model.



**Fig. 1:** Adopted reference frames

### B. Process Model and Measurement Model

For filtering purposes, the discrete process model and the measurement model of the vehicle can be described by

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \quad (2)$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (3)$$

where  $k$  is a time index,  $f(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  is a nonlinear function of the state vector  $\mathbf{x} = [x, y]$  and the input vector  $\mathbf{u}_k = [u, v, \psi]$ ,  $h(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}$  is a nonlinear function that yields the measurement taken at time  $k$  by the vehicle at a given state  $\mathbf{x}_k$ ;  $\mathbf{y}_k$  the measurement vector taken at time  $k$ ;  $\mathbf{w}_k$  is the process noise sequence represented as additive white Gaussian noise, with zero mean and constant covariance matrix  $Q$ ;  $\mathbf{v}_k$  is the measurement noise that models map errors and sensor noise, represented as additive white Gaussian noise, with zero mean and time-invariant covariance matrix  $R$ . The process and measurement noise sequences are assumed to be mutually independent. For the discrete process model used by the particle filter, introduced in Section III,  $f(\mathbf{x}_k, \mathbf{u}_k)$  is assumed to be a discretized version of (1).

### C. Bayesian Filter Approach

The nonlinear Bayesian estimation problem consists in computing recursively the probability density function (PDF) of the state  $\mathbf{x}_k$  conditioned on a given set of observations  $\mathbf{y}_{1:k-1}$  up to time step  $k-1$  and the previous states  $\mathbf{x}_{0:k-1}$ . Using the Chapman-Kolmogorov equation, the prior PDF of the state at time  $k$  can be written as

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}, \quad (4)$$

see [13]. When a new observation  $y_k$  becomes available, the prior PDF  $p(x_k | y_{1:k-1})$  is updated according to Bayes' Theorem, obtaining the posterior PDF  $p(x_k | y_{1:k})$  as

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})}, \quad (5)$$

where  $p(\mathbf{y}_k | \mathbf{y}_{1:k-1})$  is a normalization constant given by

$$p(\mathbf{y}_k | \mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) d\mathbf{x}_k. \quad (6)$$

It is now possible to obtain a point estimate  $\hat{x}^{MSSE}$  that minimizes the mean state estimation error, given by

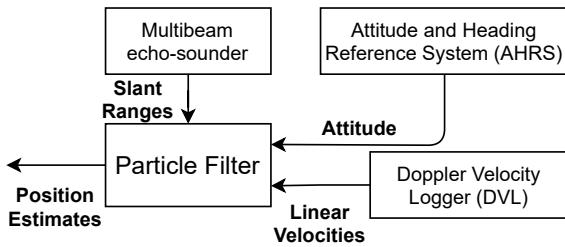
$$\hat{x}^{MSSE} = \int \mathbf{x}_k p(\mathbf{x}_k | \mathbf{y}_{1:k}) d\mathbf{x}_k \quad (7)$$

The posteriori of the state, also referred to as belief, can be expressed as

$$Bel(x_k) = p(x_k | y_{1:k}). \quad (8)$$

### III. PARTICLE FILTER

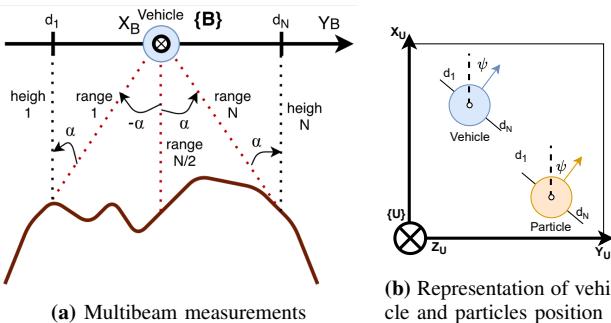
For the positioning problem, formulated in the context of Bayesian estimation theory, there is no analytical solution to the posterior PDF. For this reason, a particle filter formulation is adopted to yield a sub-optimal, non-parametric implementation of the Bayesian filter described [14]. In this method, the posterior PDF is approximated by a set of random samples with associated weights (particles), drawn from that posteriori PDF. Each particle weight represents the probability of that particle encoding the real state of the system. In the set-up adopted, the system's measurement vector,  $\mathbf{y}_k$ , consists of the processed slant ranges provided by a multibeam echo-sounder at time step  $k$ . Fig. 2 shows the set-up adopted for the particle filter implementation.



**Fig. 2:** Particle Filter experimental setup

#### A. Using the Multibeam Echosounder in a PF structure

The Multibeam echosounder sensor is used to obtain slant range measurements to the terrain. Unlike a regular sonar, this sensor emits acoustic waves in a fan-like shape. The time taken for the acoustic waves to reflect off the sea bottom coupled with beam-forming techniques allow the sensor to obtain not only ranges but also the orientation of the beams with respect to the sensor receiver [15]. In a general case, the multibeam has  $N$  beams which provide  $N$  slant range measurements at each time instant  $k$ . In order to compare the data given by this sensor with the data available in the DEM, the first step is to convert the (range, angle) measurements to (height, distance) measurements, as explained graphically in Fig. 3 a).



**Fig. 3:** Multibeam usage diagrams

Consider for each sonar beam the angle  $\alpha$  associated to the measured *range*, the conversion described above is given by

$$\begin{bmatrix} h \\ d \end{bmatrix} = \text{range} \cdot \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix}, \quad (9)$$

where  $[h, d]^T$  encodes the height and a distance expressed in the lateral y-axis of the vehicle. Applying (9) to the  $N$  measurements provided by the multibeam one must find, for each particle, how the distances  $[d_1, \dots, d_N]$  relate to  $XY$  coordinates in the bathymetric map according to Fig. 3 b). Since each particle encodes a distinct possible state of the system  $[x, y]$ , we can compute (for each particle) where each  $XY$  measurement lies in the map from the equations

$$\begin{cases} \gamma_n = \psi - 90^\circ, n < N/2 \\ \gamma_n = \psi + 90^\circ, n \geq N/2 \end{cases} \quad (10)$$

$$\begin{bmatrix} x_{m,n} \\ y_{m,n} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + d_n \begin{bmatrix} \cos(\gamma_n) \\ \sin(\gamma_n) \end{bmatrix}, n = 1, \dots, N \quad (11)$$

where  $\gamma_n$  is the angle formed between  $x_U$  and the lateral y-axis of the vehicle. The coordinates  $[x_{m,n}, y_{m,n}]$  correspond to the positions in the DEM at which the particle  $m$  believes the beam  $n$  has hit the seabed. With this coordinate it is now possible to extract the value  $h_n^m$  which corresponds to the expected height for the combination of particle  $m$  and beam  $n$ . Afterwards, this value is compared with the real height  $h_n$  computed previously according to (9).

The main advantage of this approach is its efficiency due to the fact that it is not required to swipe an entire map line to compare measurements.

#### B. Algorithm Outline

1) **Initial Particle Distribution:** In the particle filter algorithm, each particle represents a different possible state of the system. Therefore, when an initial belief of the state is available, the initial particles should follow a **Gaussian distribution** centered at the state belief. On the other hand, when a belief of the initial state of the system is not available, e.g. there is no estimate of the vehicle position, the initial particles should be drawn from an **uniform distribution** in order to achieve maximum coverage of the area of interest.

2) **Prediction:** In this step the particle sample set is propagated through the process model, meaning that for each state of generated particles in  $k$  the next state at  $k+1$  is obtained. The process noise in the motion model represents how accurate the motion model fits the actual motion of the vehicle in surge and sway.

3) **Update:** The likelihood PDF is computed based on sensor measurements. The Multibeam echosounder provides  $N$  pairs of ranges-angles at each time instant  $t_k$ , related to the sensor measurement frequency. The measurements are not used directly, but are instead converted to a set of heights-distances, as mentioned in section III-A. The filter then computes for particle  $m = 1, \dots, M$  the associated heights, via the bathymetric map, according to the previously

obtained distances. Afterwards, the likelihood is computed by comparing the associated heights on the map  $h_n^m$  with the measured heights  $h_n$  where  $n = 1, \dots, N$ , yielding

$$p(\mathbf{y}_k | \mathbf{x}_{k|k-1}^m) = \prod_{n=1}^N \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(h_n - h_n^m)^2}{2\sigma^2}}, \quad (12)$$

where  $N$  is the number of measurements available at time  $k$  and  $\sigma$  is the standard deviation of the sensor measurement errors.

**4) Resample:** Resampling consists in replicating the particles with higher weights and eliminating the particles with lower weights. There are several different resampling algorithms as mentioned in [16]. In the context of this work the *Systematic Resampling* is used, executed in  $O(N)$  time complexity and provides slightly improved particle filter estimates than other resampling algorithms. The resampling step can be heavy to run at every iteration, especially when the number of particles is high, as well as lead to particle depletion. To address this issue, the resample step occurs only when a given number of effective particles  $N_{eff}$  falls below a pre-defined threshold.

The above steps are detailed in Algorithm 1 below.

---

#### Algorithm 1 Particle Filter

---

**Initialization:** Draw  $N$  particles:

$$\mathbf{x}_0^i \sim p(\mathbf{x}_0)$$

where  $p(\mathbf{x}_0)$  is the prior probability distribution

$$w_0^i = 1/N$$

**Prediction:**

Propagate the sample set through the process model equation, yielding

$$\mathbf{x}_{k|k-1}^i = f(\mathbf{x}_{k-1|k-1}^i, \mathbf{u}_{k-1}) \sim q(\mathbf{x}_k | \mathbf{x}_{k|k-1}^i, \mathbf{y}_k)$$

**Update:**

Update the weights using the likelihood function, according to

$$w_k^i = w_{k-1}^i p(\mathbf{y}_k | \mathbf{x}_{k|k-1}^i)$$

Normalize the weights:

$$w_k^i = w_k^i / \sum_{i=1}^N w_k^i$$

Estimate the current state:

$$\hat{\mathbf{x}}_{k|k}^{MMSE} = \sum_{i=1}^N w_k^i \mathbf{x}_{k|k-1}^i$$

Compute the effective sample size:

$$N_{eff} = 1 / \sum_{i=1}^N (w_k^i)^2 \quad (13)$$

Resample if  $N_{eff} \leq N_{thresh}$  and reset the weights  
**Go to Prediction**

---

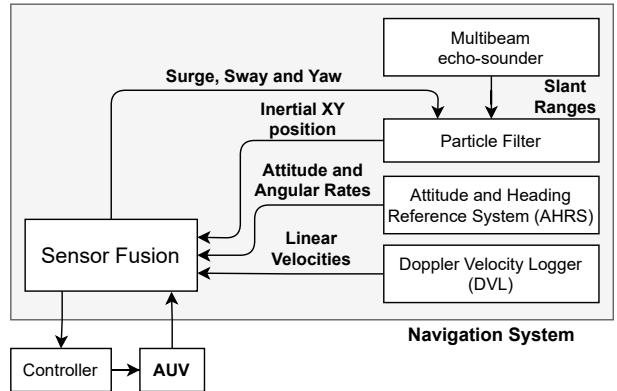
#### IV. SENSOR FUSION

At this stage, the main goal is to integrate the PF introduced in Section III into the navigation system of MEDUSA.

At the core of the MEDUSA navigation system is a sensor fusion framework that combines sensor measurements to produce estimates of unknown vehicle variables. This framework is built around a Kalman filter (KF) [17], which provides a solution to many tracking and data prediction tasks and finds excellent use in control, navigation and guidance.

In this architecture, the navigation system interprets all the data provided by a suite of onboard sensors (DVL, AHRS, etc) and the particle filter estimates as measurements. The navigation system provides estimates of the pose, velocity and acceleration of the vehicle. Additionally, the velocity and yaw estimates will be fed back into the particle filter to provide the position estimates of the vehicle. Therefore, the strategy was to create a feedback loop between the particle filter and the Kalman filter as shown in Fig. 4.

The vehicle is also equipped with control algorithms described in detail in [18].



**Fig. 4:** Implementation architecture of the closed loop system with navigation and control systems

As it will become clear later, in the set-up adopted, the navigation filter design model is linear. For this reason, we adopt a linear Kalman filter structure.

##### A. Kalman Filter outline

The algorithm works in a two-step process. In the *predict* step, the KF looks one-step forward in time to predict the estimates of the current state vector and state covariance using an inner dynamic model. In the second step, known as *update*, the KF gain is computed using the state covariances and measurement covariance. This gain is then used to update the estimated state of the vehicle.

In the KF,  $\mathbf{x}$  and  $\mathbf{A}_k$  represent the state vector and transition matrix, respectively. The vector  $\mathbf{u}$  and matrix  $\mathbf{B}_k$  represent the input vector and model matrix, respectively.  $\mathbf{P}_{k|k}$  represents

the covariance matrix of the a posteriori estimate, and  $\mathbf{Q}_k$  is the process noise covariance matrix, with the noise assumed to follow a zero-mean Gaussian distribution.

The vector  $\tilde{\mathbf{y}}$  denotes the error between the predicted state output and the measured state output,  $\mathbf{S}_k$  represents the innovation covariance matrix,  $\mathbf{R}_k$  is the measurement covariance matrix, with the measurement noise also assumed to follow a zero-mean Gaussian distribution,  $\mathbf{K}_k$  represents the optimal Kalman gain of the filter and  $\mathbf{C}_k$  denotes the observation matrix of the state. The variable  $\mathbf{z}_k$  is the measurement vector.

The Kalman Filter is summarized in Algorithm 2 below.

---

**Algorithm 2** Kalman Filter

---

**Initialization:**

$$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$$

$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T(\mathbf{x}_0 - \hat{\mathbf{x}}_0)]$$

**Prediction:**

Propagate the state estimate through the process model equation,

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$$

Compute the covariance matrix,

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^T + \mathbf{Q}_k$$

**Update:**

Compute the innovation

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{C}_k \mathbf{x}_{k|k-1}$$

Compute the innovation covariance

$$\mathbf{S}_k = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R}_k$$

Optimal Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \mathbf{S}_k^{-1}$$

Update the a posteriori state estimate:

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k \mathbf{y}_k$$

Update the a posteriori state covariance:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

---

**Go to Prediction**


---

Unlike the process model adopted for the particle filter, a linear motion model was adopted in the Kalman filter. In this model, the vehicle is assumed to move with approximately constant acceleration and angular rate. Every filter input is interpreted as a sensor measurement. Therefore, there are no process model inputs,  $\mathbf{u}$ , to this system which results in the process input matrix  $\mathbf{B} = [\mathbf{0}]$ .

The key advantage of this model, especially when comparing with more typical filtering structures, such as complementary filters, is the flexibility it provides. By considering every input of the filter as a measurement, it facilitates the combination of several sensors that can provide measurements of the same variables. For example, if a USBL system were available,

it would be possible to fuse both position measurements provided by the particle filter and the USBL. This flexibility comes at the cost of added complexity when it comes to tuning the navigation filter.

### B. Kalman Filter Process Model

The filter design model captures, in a simple manner, the evolution of the origin of the body frame attached to the vehicle in 2D. The relevant states variables are its position and linear velocity. We also include in the model the evolution of yaw and yaw rate, included in vector  $\mathbf{x}$ . The resulting state is given by

$$\mathbf{x} = [x \quad y \quad \dot{x} \quad \dot{y} \quad \ddot{x} \quad \ddot{y} \quad \psi \quad \dot{\psi}]^T \quad (14)$$

and the corresponding state transition matrix is

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \delta t & 0 & 0.5\delta t^2 & 0 & 0 & 0 \\ 0 & 1 & 0 & \delta t & 0 & 0.5\delta t^2 & 0 & 0 \\ 0 & 0 & 1 & 0 & \delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (15)$$

where  $\delta$  represents the prediction period of the filter.

### C. Kalman Filter Observation Model

The observation model of the Kalman Filter represents the relation between the variables measured by the available sensors and the state vector. In our case, the measurement vector  $\mathbf{z}$  and the observation matrix  $C$  are given by

$$\mathbf{z} = [x_{PF} \quad y_{PF} \quad \dot{x}_{DVL} \quad \dot{y}_{DVL} \quad \psi_{AHRS} \quad \dot{\psi}_{AHRS}]^T \quad (16)$$

and

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & \delta t & 0 & 0.5\delta t^2 & 0 & 0 & 0 \\ 0 & 1 & 0 & \delta t & 0 & 0.5\delta t^2 & 0 & 0 \\ 0 & 0 & 1 & 0 & \delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (17)$$

In the above,  $x_{PF}$  and  $y_{PF}$  are the position measurements provided by the particle filter,  $\dot{x}_{DVL}$  and  $\dot{y}_{DVL}$  the velocity provided by the DVL and  $\psi_{AHRS}$  and  $\dot{\psi}_{AHRS}$  the yaw and yaw-rate provided by the AHRS. Note that velocity measured in the body frame  $\{B\}$ , such as surge and sway measured by the DVL, can be expressed in the inertial frame  $\{U\}$  using (1), before being incorporated in the measurement vector  $\mathbf{z}$ . This relation is also relevant when applying conversions from  $\{U\}$  to  $\{B\}$ .

#### D. Outlier Detection and Rejection

Additionally, the filter can be tuned to reject certain state predictions when the error surpasses a pre-defined threshold outlier rejection parameter. This is simply done by defining a normalized error consisting in scaling the squared state output error by the innovation factor, as follows

$$e_n = \tilde{\mathbf{y}}_k^T S_k^{-1} \tilde{\mathbf{y}}_k \quad (18)$$

and cut off any predictions that result in an error below a certain threshold  $e_n < th$ .

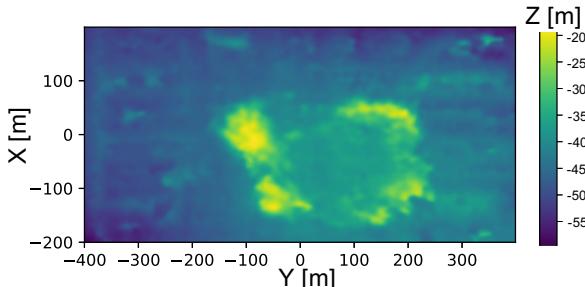
## V. SIMULATION RESULTS

The performance of the proposed particle filter and the resulting navigation system are assessed within the UUV simulator [9] using two separate approaches. First, the particle filter is used as an individual localization solution to the Rexrov AUV performing a lawnmower trajectory, that is, the filter receives direct measurements from the multibeam, DVL and AHRS, and its performance is compared against that obtained with a standard DR technique. In the second and final approach, the particle filter is incorporated in the sensor fusion framework of the MEDUSA AUV navigation system. Finally, the performance of the complete system is analyzed and its results are compared with the ones provided by DR. In both approaches the performance is evaluated with multiple simulation runs, where the differences between position estimates and ground truth position of the vehicle are computed in terms of the root-mean-square-error (RMSE).

#### A. Prior Bathymetric Maps

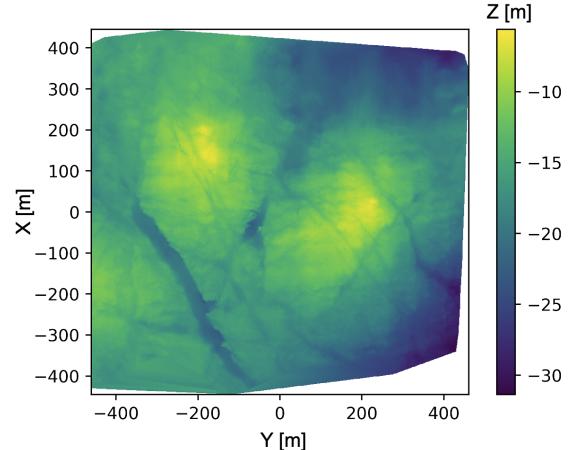
The bathymetric maps used for carrying out the simulation and validation of results were acquired *a priori* at two different sites:

- Dom João de Castro, Portugal shown in Fig. 5. This region is an isolated seamount located in the Azores archipelago, between the islands Terceira and S. Miguel.



**Fig. 5:** Bathymetric map data from the Azores region.

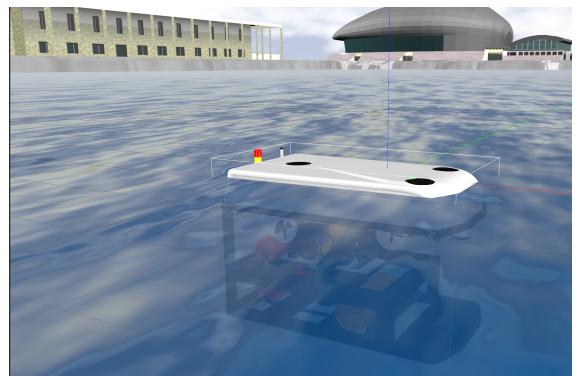
- Zhongsha Reef, Qingdao, China shown in Fig. 6. The region is predominantly flat with a low number of conspicuous features.



**Fig. 6:** Bathymetric map data from Zhongsha Reef region in China.

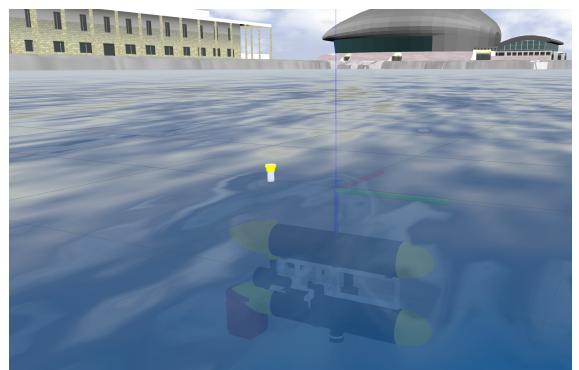
#### B. Simulation Environment

In order to conduct the simulations in a realistic manner, the bathymetric maps presented in the previous section were used to generate 3D scenarios of the environment. Furthermore, there was also the necessity to incorporate the vehicles 3D models and dynamics in these scenarios. The particle filter standalone simulation was implemented in the default autonomous underwater vehicle provided by the simulation environment - the Rexrov - as shown in Fig. 7.



**Fig. 7:** Rexrov in simulated gazebo world

To generate the results for sensor fusion, the MEDUSA class of vehicles was integrated in the 3D simulator, Fig. 8.



**Fig. 8:** MEDUSA in simulated gazebo world

### C. Particle Filter results

The parameters of the particle filter used in the simulations are given in Table I. A large covariance of the measurement noise was adopted in order to indirectly account for errors that plague underwater environments.

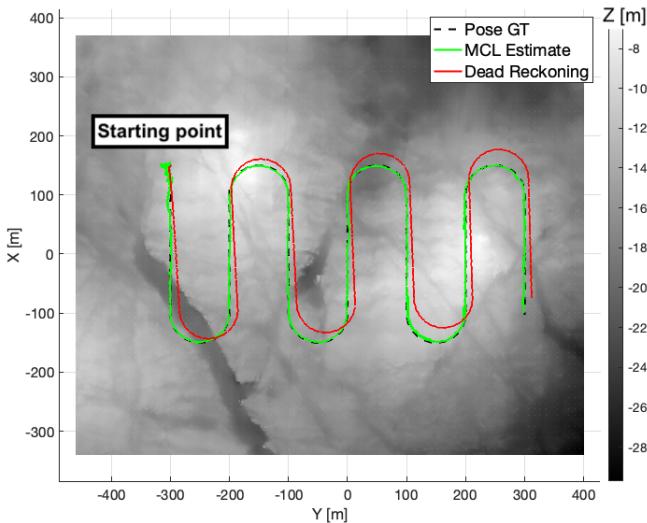
**TABLE I:** Filter parameters

Parameters	Values
Update frequency, $T^{-1}$	10 Hz
Number of particles, $N$	1000
Resample threshold $N_{thresh}$	650
Std. dev. for measurement model, $\sigma_\gamma$	10.0 m
Process covariance	$\begin{bmatrix} 0.5 & 0.00 \\ 0.00 & 0.5 \end{bmatrix}$ m
Percentage of particles used in the estimation	0.05

The standard deviation of the measurement noise affecting the DVL readings is characterized by  $\sigma = 0.01$ m/s. The standard deviation considered for the yaw rate given by the AHRS is  $\sigma_r = 0.034$ rad/s. The Multibeam echosounder is characterized by having a minimum range of 1m and a maximum of 100m, with range measurement errors given by  $\sigma = 0.023$ m. The sweeping angles are  $[-\pi/4, \pi/4]$  rad. For this experience, a lawn mowing trajectory was performed by the vehicle.

The positions of the filter particles can be initialized with a uniform distribution along a rectangular section of the map containing the vehicle, or with a Gaussian distribution whose parameters approximate an initial guess of the vehicle position on the map. The two initialization methods are chosen previously according to the mission planning and scenario.

For the first experiment, the filter was tested using a map of the region of Zhongsha Reef, China, as shown in Fig. 9.

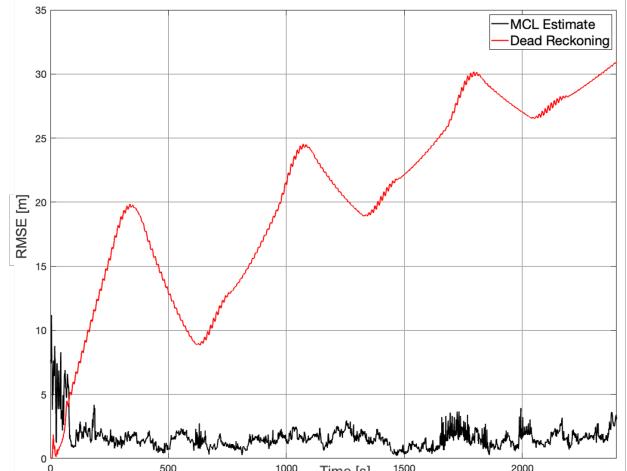


**Fig. 9:** Real, DR and estimated X-Y trajectory in the region of China.

The particles were initialized uniformly distributed over a compact interval with coordinates  $x \in [120, 180]$ m and  $y \in$

$[-270, 330]$ m. The pose ground-truth of the vehicle is denoted "Pose GT" while the position estimate provided by the particle filter is denoted "MCL Estimate".

Fig. 10 shows a plot of the Root Mean Square Error (RMSE) of the vehicle position estimates obtained after 15 simulation runs of the same mission scenario. The position errors are bounded by approximately 5 meters, while the dead-reckoning error grows unbounded. Note that, initially, the particle filter position estimate error is larger than the dead-reckoning method as a result of initializing the DR system with the ground truth position, which is not realistic but adds weight to the fact that the particle filter recovers well and exhibits good performance.



**Fig. 10:** RMSE of position estimates for 15 runs in the China map region.

Additionally, it is possible to observe oscillations in the RMSE plot regarding the DR error. These can be easily explained by the existence of non negligible sway motion of the Rexrov vehicle control system and also the integration of the bias from the AHRS sensor. In fact, the AHRS model used in simulation includes a sensor bias that follow a random walk with zero-mean white Gaussian noise [9]. Conjugated with the performed lawn mowing path, this translates to certain locations on the map where the positioning error increases and decreases, as shown in Fig. 9.

### D. Sensor Fusion Results

The performance of the proposed sensor fusion navigation system was assessed in simulation using the MEDUSA vehicle. The PF parameters remain the same as in Section V-C with the particles initially distributed uniformly in a square of 50 meters in side, centered at a point where the vehicle position is believed to be located.

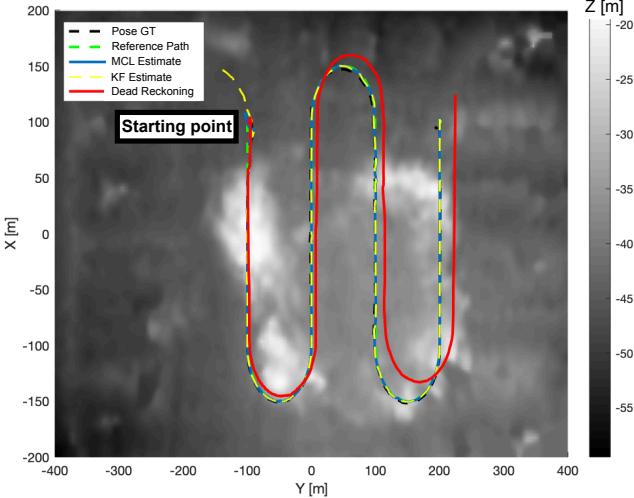
The parameters of the Kalman Filter used for sensor fusion are given in table II. In this setup, the standard deviation of the PF position estimates is computed based on the standard deviation of the particles themselves with respect to 5% of particles with the highest attributed weight.

**TABLE II:** Kalman Filter parameters

Parameters	Values
Update frequency, $T^{-1}$	10 Hz
Std. dev of position, $[x, y]$	0.14m
Std. dev of translation, $[\dot{x}, \dot{y}]$	0.14m/s
Std. dev of acceleration, $[\ddot{x}, \ddot{y}]$	0.9m/s <sup>2</sup>
Std. dev of orientation, $\psi$	0.228rad
Std. dev of orientation rate, $\dot{\psi}$	3.16rad/s
Std. dev of DVL measurements (expressed in inertial frame)	0.1m/s
Std. dev of AHRS angle measurements,	0.03rad
Std. dev of AHRS angular rate measurements,	0.001rad/s
Std. dev of PF position measurements	Dynamic
Outlier rejection threshold, $th$	16

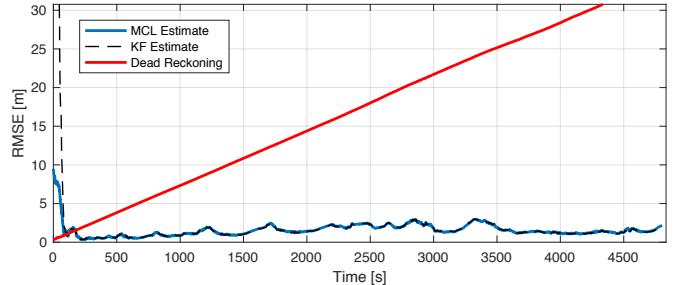
It is important to emphasize that the results introduced in this section result from the closed loop feedback between the control systems implemented in the MEDUSA control stack and the sensor fusion approach, unlike the previous experiments performed with the Rexrov vehicle, where the positioning system was running independently of the control system. In particular, a path following control law was implemented to make the vehicle follow a desired reference path with a given speed profile along the path [18].

For the experiment in Fig. 11 a new simulation was performed in the Azores region. Following the same evaluation as in Section V-C, 15 simulation runs were performed in order to generate the RMSE plots shown in Fig. 12. The sensor fusion estimated position is denoted "KF estimate".



**Fig. 11:** Real, DR , KF and MCL X-Y trajectory in the region of Azores

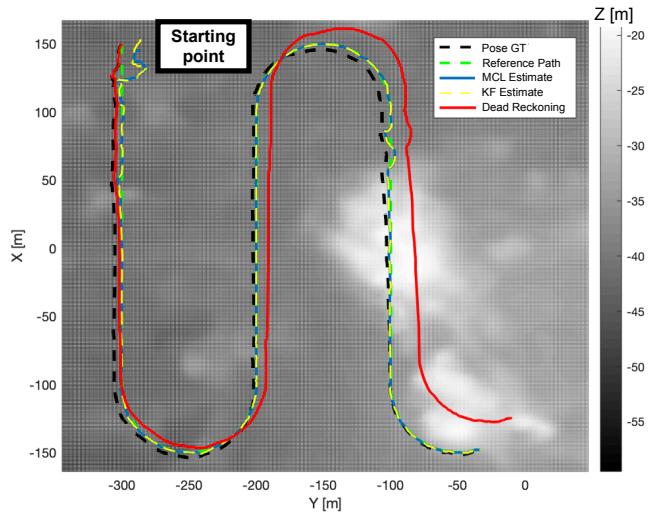
Initially, the PF outperforms the KF accuracy as a result of the initiation step performed. The KF is initialized at an arbitrary position inside the square of 50m also used for the initialization of the particle filter. Due to high covariance in the initial iterations of the PF, the sensor fusion rejects most of the position measurements provided by the PF. As a result, the RMSE is higher at the beginning since the PF is the only available methodology for estimating the vehicle's position.



**Fig. 12:** RMSE region of Azores

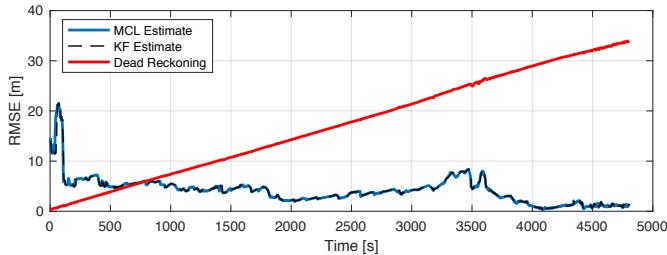
However, almost immediately after the initialization, the KF RMSE reduces as the covariance of the PF starts to decrease. From then on, the KF position estimate closely follows the PF position estimate, as it is the lowest covariance estimator for the position available to the navigation system. The RMSE remains bounded by 4 meters whereas the DR RMSE grows unbounded.

The results above represent the behavior of the PF and KF estimation when performing a mission in the region of the map with significant variations of elevation, which is valuable in geophysical based navigation. Nevertheless, it is also important to evaluate the robustness of this system in regions of smooth variations in elevation, that is, with a reduced number of conspicuous features. For this purpose, a new lawnmower trajectory was performed, as shown in Fig.13, in a region with low gradient in elevation, represented by the color gradient.



**Fig. 13:** No gradient X-Y trajectory in the region of Azores

The correspondent plots of the RMSE are presented in Fig. 14. As expected, when compared to the results obtained in the previous scenario, represented in Fig.12, the KF performance deteriorates in these type of regions with low features. However, the RMSE still outperforms the DR RMSE, maintaining an error bound of 10 meters. It is also possible to observe the RMSE error decreasing when the vehicle goes over certain regions where the gradient increases.



**Fig. 14:** No gradient RMSE region of Azores

Moreover, using the sensor fusion as the core of the navigation system enhances the quality of the filtered measures. For example, by comparing the DR RMSE plots from the standalone implementation, shown in Fig. 10, and from the sensor fusion, shown in Fig. 12, it is observed that the latter does not show any type of oscillation, using the same suite of sensors. The key difference is that the sensor fusion methodology filters the existing bias from the set of sensors used in the navigation system, such as the one from the AHRS. Implementing the standalone methodology to estimate the vehicle position, the DR integrated direct measures from the sensors, therefore integrating the constant bias, as previously explained in Section III. On the other hand, implementing the sensor fusion, the DR integrated direct measures from the KF, which filtered the bias, resulting in no oscillations from the DR RMSE. Furthermore, this approach enables the use of the filtered sensor data by the control systems of the vehicle.

## VI. MAIN CONCLUSIONS

The problem of vehicle underwater localization by measurement of bathymetric data from a priori information of the surrounding environment was addressed in this paper. A Bayesian problem was formulated to estimate the position of the vehicle in an horizontal plane using slant range measurements provided by a Multibeam echosounder. The proposed solution to this problem is a filtering algorithm included in the class of sequential Bayesian estimators designated as particle filters, whose performance is tested individually assuming true readings from a suite of sensors that enable estimation of vehicle position, outperforming conventional methods of navigation such as dead-reckoning.

Motivated by the achieved performance of the particle filter, the solution is included in a navigation system implementation of a Kalman filter that fuses information about the vehicle state. The sensor fusion approach alongside the particle filter solution achieved good performance at estimating the vehicle position, with a bounded error of 10 meters in worst case scenarios, validating the use of the particle filter in a sensor fusion methodology towards increasing filtering performance. Furthermore, the presented architecture can be extended for geomagnetic based navigation. Future work will include an extension of the problem to the 3D case, with a more detailed model of the vehicle motion that takes into consideration unknown ocean currents. Moreover real water trials should be conducted in order to access the performance of the

navigation system in a real world environment.

## REFERENCES

- [1] A. Gómez-Espinosa, E. Cuan-Urquiza, J. González-García. (2020). *Autonomous Underwater Vehicles: Localization, Navigation, and Communication for Collaborative Missions*. Applied Sciences. 10. 1256.
- [2] Pedro Batista, Carlos Silvestre, Paulo Oliveira, *A Sensor-based Long Baseline Position and Velocity Navigation Filter for Underwater Vehicles*, IFAC Proceedings Volumes, Volume 43, Issue 14, 2010, pp 302-307.
- [3] M. Morgado, P. Batista, P. Oliveira, C. Silvestre, *Position and Velocity USBL/IMU Sensor-based Navigation Filter*, IFAC Proceedings Volumes, Volume 44, Issue 1, 2011, Pages 13642-13647.
- [4] Nguyen T. Hung, N. Crasta, David Moreno-Salinas, António M. Pascoal, Tor A. Johansen, *Range-based target localization and pursuit with autonomous vehicles: An approach using posterior CRLB and model predictive control*, Robotics and Autonomous Systems, Volume 132, 2020.
- [5] Meduna, D.K. *Terrain Relative Navigation for Sensor-Limited Systems with Applications to Underwater Vehicles*, Phd Thesis, Standford University. Standford, CA, USA, 2011.
- [6] J. Quintas, J. Cruz, A. Pascoal and F. C. Teixeira, *A Comparison of Nonlinear Filters for Underwater Geomagnetic Navigation*, 2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV), 2020, pp. 1-6.
- [7] T. Nicosevici, R. Garcia, M. Carreras and M. Villanueva, *A review of sensor fusion techniques for underwater vehicle navigation*, Oceans '04 MTS/IEEE Techno-Ocean '04 (IEEE Cat. No.04CH37600), 2004, pp. 1600-1605 Vol.3.
- [8] A. Cavallo, A. Cirillo, P. Cirillo, G. De Maria, P. Falco, C. Natale, S. Pirozzi, *Experimental Comparison of Sensor Fusion Algorithms for Attitude Estimation*, IFAC Proceedings Volumes, Volume 47, Issue 3, 2014, Pages 7585-7591.
- [9] Musa Morena Marcusso Manhães et al., *UUU Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation*, OCEANS 2016 MTS/IEEE Monterey, September, 2016.
- [10] N. Koenig and A. Howard. *Design and use paradigms for Gazebo, an open-source multi-robot simulator*, 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), vol.3, 2004
- [11] Berg, Viktor. *Development and Commissioning of a DP system for ROV SF 30k*. MS thesis. Institutt for marin teknikk, 2012
- [12] P. C. Abreu, J. Botelho, P. Góis, A. Pascoal, J. Ribeiro, M. Ribeiro, M. Rufino, L. Sebastião and H. Silva, "The MEDUSA class of autonomous marine vehicles and their role in EU projects", 2016.
- [13] F. C. Teixeira, J. Quintas, P. Maurya, and A. Pascoal, *Robust particle filter formulations with application to terrain-aided navigation*. International Journal of Adaptive Control and Signal Processing, 31(4):608–651, 2017.
- [14] F. Teixeira, J. Quintas, P. Maurya and A. Pascoal, *Robust particle filter formulations with application to terrain-aided navigation: ROBUST PARTICLE FILTERS FOR TERRAIN-AIDED NAVIGATION*, International Journal of Adaptive Control and Signal Processing, Volume 31, 2016
- [15] A. Rønholde, L. Yang, T. Taxt, S. Holm. *High-resolution beamforming for multibeam echo sounders using raw EM3000 data*. 2. 923 - 930 vol.2. 10.1109/OCEANS.1999.804997.
- [16] Hol, Jeroen and Schön, Thomas and Gustafsson, Fredrik. *sOn Resampling Algorithms for Particle Filters*, 2006, IEEE Nonlinear Statistical Signal Processing Workshop, pp. 79 - 82.
- [17] Rhudy, Matthew B and Salguero, Roger A and Holappa, Keaton, *A Kalman Filtering Tutorial for Undergraduate Students*, International Journal of Computer Science & Engineering Survey, Feb., 2017
- [18] F. Vanni, A. Aguiar, A. Pascoal, *Cooperative Path-Following of Underactuated Autonomous Marine Vehicles with Logic-based Communication*, IFAC Proceedings Volumes, Volume 41, Issue 1, 2008, Pages 107-112