





Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

# Range-only single-beacon tracking of underwater targets from an autonomous vehicle: from theory to practice

IVAN MASMITJA<sup>1</sup> , (Member, IEEE), SPARTACUS GOMARIZ<sup>1</sup> , (Member, IEEE), JOAQUIN DEL-RIO<sup>1</sup> , (Member, IEEE) BRIAN KIEFT<sup>2</sup>, TOM O'REILLY<sup>2</sup>, PIERRE-JEAN BOUVET<sup>3</sup> , (Member, IEEE) AND JACOPO AGUZZI<sup>4</sup>

<sup>1</sup>SARTI Research Group, Electronics Department, Universitat Politècnica de Catalunya, Barcelona, Spain

<sup>2</sup>Monterey Bay Aquarium Research Institute (MBARI), California, USA

<sup>3</sup>Underwater Acoustics Lab. ISEN Brest Yncrea Ouest, France

<sup>4</sup>Marine Science Institute (ICM). Consejo superior de Investigaciones Científicas (CSIC), Barcelona, Spain

Corresponding author: Ivan Masmitja (e-mail: ivan.masmitja@upc.edu).

This work was partially supported by the project JERICO-NEXT from the EC-H2020 (grant agreement number 654410; EvoLUL TNA sub-project). We are also grateful for the financial support from the Spanish Ministerio de Economía y Competitividad (contract TEC2017-87861-R project RESBIO), and by Generalitat de Catalunya 'Sistemas de Adquisición Remota de datos y Tratamiento de la Información en el Medio Marino (SARTI-MAR)' 2017 SGR 376. The main author of this work has a scholarship (FPI-UPC) from Universitat Politècnica de Catalunya (UPC) for his PhD research (agreement number 175/2015).

**ABSTRACT** Underwater localisation is one of the main problems which must be addressed in subsea exploration, where no Global Positioning System (GPS) is available. In addition to the traditional underwater localisation systems, such as Long Base Line (LBL), new methods have been developed to increase navigation performance, flexibility, and to reduce deployment costs. For example, Range-Only and Single-Beacon (ROSB) is based on an autonomous vehicle which localises and tracks different underwater targets using slant range measurements carried out with acoustic modems. This paper presents different strategies to improve the ROSB tracking methods. The ROSB target tracking method can be seen as a Hidden Markov Model (HMM) problem. Using Bayes' rule, the probability distribution function of the HMM states can be solved by using different filtering methods. Here, we present and compare different methods under different scenarios, both evaluated in simulations and field tests. The main mathematical notation and performance of each algorithm are presented, where best practice has been derived. From a methodological point of view, this work advanced the understanding of accuracy that can be achieved by using ROSB target tracking methods with autonomous underwater vehicles.

**INDEX TERMS** Particle filter, range-only target tracking, single-beacon, autonomous underwater vehicles, acoustic modems, slant range.

## I. INTRODUCTION

UNDERWATER localisation is one of the main problems which has to be addressed in ocean exploration, where no GPS is available due to the high attenuation that electromagnetic waves suffer in seawater [1]. Therefore, most underwater positioning systems have to be conducted with acoustic communications, despite the channel limitations, such as frequency dependent attenuation, Doppler spread and multipath propagation [2].

One of the first acoustic underwater localisation methods was the LBL, which appears in the 1960s and 1970s [3]. Since then, different alternatives have been developed, such

as the Short Base Line (SBL), the Ultra-Short BaseLine (USBL), or the GPS Intelligent Buoys (GIB) systems [4] and [5]. In addition to these methods, new techniques are being developed based on the Range-Only and Single-Beacon (ROSB). For example, the moving long baseline [6] or the cooperative range-only tracking methods [7], which use the improvements in autonomous vehicles' performance, and their capabilities to work in more complex scenarios (e.g. [8]).

In general, the ROSB methods are based on an autonomous vehicle which is used as a tracker (or observer). This vehicle conducts a set of manoeuvres in order to track (or localise)

some target(s). In this manoeuvre, the vehicle periodically performs new slant range measurements using the Time Of Flight (TOF) of exchanged messages between the tracker and the target (e.g. [6]), whereas the LBL method uses the Time Difference Of Arrival (TDOA) between different well localized and synchronized transponders deployed previously on the seafloor (e.g. [9] [10] [11]).

The interest in ROSB has increased in recent years as a consequence of the necessity to reduce localisation costs (e.g. transponders' deployment and clocks' synchronization) [12] [13], or to increase the vehicles' autonomy capabilities in complex missions [8] [14].

The ROSB target tracking methods can be studied from different points of view. For example, the tracker's optimal path, where [13] and [15] developed a complete analytical study for an optimal sensor placement in an underwater target localisation scenario. In [6] the optimal path for static target localisation was not only demonstrated analytically and using simulations, but also in real field tests. In both cases a Least Square (LS) and a Maximum Likelihood (ML) estimator where used, and a circumference path was determined as optimal. Later, in [7], the authors derived the same idea but for cooperative range-based underwater moving target localisation. In that case an Extended Kalman Filter (EKF) estimator was used.

On the other hand, if system observability is not taken into account, the algorithms to estimate the target's position have to solve complex multimodal probability density functions, yielding more complex algorithms. For example, in [16] the authors studied and compared the Particle Filter (PF) and the batch-MAP estimators. However, they considered an indoor scenario where a robot with movement constraints tracked a second one.

The ROSB target tracking method can also be seen as a HMM problem. Generally, the HMM is defined as a sequence of states, known as a Markov chain, and a set of observations for each state [17]. Using Bayes' rule

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

the probability distribution function of the HMM states can be derived given a set of observations  $\mathbf{z} \in \mathbb{R}^m$ , and therefore, the current state  $\mathbf{x} \in \mathbb{R}^{2n}$  can be estimated. Where  $m$  indicates the number of observations carried out, and  $n$  can be either 2 or 3, which is the space dimension of the problem. And  $p(\mathbf{x}_k|\mathbf{z})$  is the posterior probability distribution, expressed also as  $p(\mathbf{x}_k|\mathbf{z}_{:k})$ ;  $_{:k}$  subscript denotes all observations up to  $k$ . The  $p(\mathbf{x}_{k-1})$  is the prior probability distribution expressed also as  $p(\mathbf{x}_k|\mathbf{z}_{:k-1})$ . And finally,  $p(\mathbf{z})$  is the total probability of  $\mathbf{z}$  [18], expressed also as  $\int_{\mathbf{x}_k} p(\mathbf{z}|\mathbf{x}_k)p(\mathbf{x}_{k-1})d\mathbf{x}_k$ , which is used as a normalized factor. However, to compute the predicted state  $\mathbf{x}_k$ , the total probability  $p(\mathbf{z})$  can be ignored, which yields in the optimal solution of the following maximization problem

$$\mathbf{x}_k^{OPT} = \underset{\mathbf{x}_k}{\operatorname{argmax}} p(\mathbf{x}_k|\mathbf{z}_{:k}). \quad (2)$$

In prediction theory and filtering, the posterior distribution can be computed recursively from the prior distribution using a prediction step  $p(\mathbf{x}_k|\mathbf{z}_{:k-1})$  and an update step  $p(\mathbf{x}_k|\mathbf{z}_{:k})$ .

In general, the existing filtering methodologies compute either the predictions with respect to the conditional probability distribution  $p(\mathbf{x}_k|\mathbf{z}_{:k})$ , such as PF, or with respect to the probability joint distribution  $p(\mathbf{x}_k, \mathbf{z}_k|\mathbf{z}_{:k-1})$ , such as EKF, see [19] and the references therein. One of the differences between these methods is the computational cost. Whereas the computational cost of the first methodology increases exponentially with the state dimension, the second one increases linearly with the state dimensions, therefore in areas with either high state dimensions or computational restrictions, this performance should be taken into consideration.

Hereinafter, the following considerations and parameters will be considered in all filtering methodologies which have been studied. To simplify the notation, a surface or underwater 2D scenario is used, where the tracker conducts manoeuvres on the sea surface to predict the target's position. This is a common procedure in cases where the target's depth can be determined with high accuracy using cheap devices (e.g. used in GPS Intelligent Buoys [20, Chapter 3]), and therefore, a 3D scenario can be projected into a 2D plane. Consequently, the state vector used for both tracker and target is defined as

$$\mathbf{x} = [x \ \dot{x} \ y \ \dot{y}]^T, \quad (3)$$

where  $x$  and  $y$  are the positions in the 2D plane, and  $\dot{x}$  and  $\dot{y}$  are their associated velocities. Finally, the observation measurement vector is defined as

$$\mathbf{z} = [z_1, \dots, z_m]^T, \quad (4)$$

where  $m$  denotes the number of observations conducted. In ROSB methods, those are the ranges between the tracker and the target, which will be computed using the slant range measured by acoustic modems and the target's depth provided by a pressure sensor during the exchange message procedure conducted to measure the range between both devices or by the prior knowledge of the target's depth.

According to the premises above, the remaining part of this work is structured as follows. In Section II different ROSB methods are presented. In Section III a brief introduction to using system observability to derive the optimal tracker's path is discussed. A set of simulations to characterize and compare different ROSB methods are presented in Section IV. Different field tests are shown in Section V. Finally, discussion and conclusions are addressed in Sections VI and VII, respectively.

## II. RANGE-ONLY AND SINGLE-BEACON METHODS

In [6] we studied the optimal path shape which should be performed by an autonomous vehicle to increase the accuracy of a static target localisation using a ROSB method. The LS and the ML estimators were compared to Cramér-Rao Bound (CRB) and different field tests. These two methods are

commonly used when no straightforward solution is possible [21]. For example, for either non-linear, non-smooth, or overdetermined systems (when  $m > n + 1$ ).

However, when the target to be localised is not static, but moving, and active tracking is desired, the LS and ML estimators are not suitable. These dynamic scenarios are typically modelled in a state-space representation of HMM, where the next state only depends on the current state, and the current measurement depends only on the current state.

In this paper, different filters and methods have been studied and compared, presenting main aspects and some implementation. These methods are:

- Extended Kalman Filter (EKF)
- Unscented Kalman Filter (UKF)
- Maximum A Posteriori (MAP) Estimation
- Particle Filter (PF)

The main aspects of these filters are presented below with a description for their implementation. We have conducted a study of the performance of different ROSB methods through simulations, and validated them through field tests. This extended study has been carried out focusing on performance comparison among different algorithms (EKF, UKF, MAP, and PF), specifically designed for 3 typical underwater scenarios: localising a static target, tracking a dynamic target, and multi-target tracking. For a methodology point of view, this work advanced the understanding of accuracy that can be achievable using ROSB localisation methods and an autonomous vehicle.

The notation employed to develop these algorithms is summarized in Table 1.

TABLE 1. Nomenclature

List of symbols	
$\mathbf{x} \in \mathbb{R}^{2n}$	State vector
$\hat{\mathbf{x}}_k \in \mathbb{R}^{2n}$	Target state vector estimated at time $k$
$\mathbf{q} \in \mathbb{R}^n$	Target position
$\mathbf{p} \in \mathbb{R}^n$	Tracker position
$\mathbf{z} \in \mathbb{R}^m$	Vector of ranges
$\mathbf{F} \in \mathbb{R}^{n \times n}$	State transition matrix
$\mathbf{Q} \in \mathbb{R}^{n \times n}$	Process noise matrix
$\mathbf{R} \in \mathbb{R}^{n \times n}$	Range measurement error covariance matrix
$\mathbf{P} \in \mathbb{R}^{n \times n}$	State covariance matrix
$h(\cdot)$	Measurement model function
$\mathbf{H} \in \mathbb{R}^n$	Jacobian matrix of $h(\cdot)$
$n \in \{2, 3\}$	Dimension of estimation problem
$m \in \mathbb{N}$	Number of measurements

### A. EXTENDED KALMAN FILTER

The EKF is the classical inference method for non-linear dynamic systems, which is based on the linearisation of the state and measurement equations along the trajectory [22] and [23]. This deterministic and parametric method estimates the target position based on the probability joint distribution as follows. First of all, the state vector of the target at

time-step  $k$  is defined by  $\mathbf{x}_k = [x_{Tk} \ \dot{x}_{Tk} \ y_{Tk} \ \dot{y}_{Tk}]^T$ . Then, assuming a constant target velocity, which is a general consideration, the motion model of the target is

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{Q}_{k-1}, \quad (5)$$

where  $\mathbf{F}$  is the state transition matrix, and  $\mathbf{Q}$  is the process noise, which has variance  $\sigma_v^2$ . Both are related to time-step  $\Delta t$ , and are described as

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

and

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 & 0 & 0 \\ \frac{1}{2}\Delta t^3 & \Delta t^2 & 0 & 0 \\ 0 & 0 & \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 \\ 0 & 0 & \frac{1}{2}\Delta t^3 & \Delta t^2 \end{bmatrix} \sigma_v^2. \quad (7)$$

On the other hand, the measurement model used at time-step  $k$  can be described by

$$\begin{aligned} h(\mathbf{x}_k) &= \|\mathbf{q}_k - \mathbf{p}_k\| + w_k \\ &= \sqrt{(x_{qk} - x_{pk})^2 + (y_{qk} - y_{pk})^2} + w_k, \end{aligned} \quad (8)$$

where  $\mathbf{q}_k \in \mathbb{R}^2$  and  $\mathbf{p}_k \in \mathbb{R}^2$  are the target and observer positions respectively in a 2D scenario, and  $w_k \sim \mathcal{N}(0, \sigma_{wk}^2)$  is a zero-mean Gaussian noise, leading to a covariance matrix equal to  $\mathbf{R} = \text{diag}[\sigma_{wk}^2]$ . Finally, the Jacobian matrix of  $h(\mathbf{x}_k)$  is computed as

$$\mathbf{H} = \left. \frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}} \right|_{\mathbf{x}_k} = \begin{bmatrix} \frac{(x_{qk} - x_{pk})}{\sqrt{(x_{qk} - x_{pk})^2 + (y_{qk} - y_{pk})^2}} \\ 0 \\ \frac{(y_{qk} - y_{pk})}{\sqrt{(x_{qk} - x_{pk})^2 + (y_{qk} - y_{pk})^2}} \\ 0 \end{bmatrix}. \quad (9)$$

Algorithm 1 has been designed to track an underwater target using the ROSB method and the EKF, which has been derived using the equations explained above, where the target state estimation  $\hat{\mathbf{x}}_k$  and its associated covariance  $\mathbf{P}_k$  at each step  $k$  are given.

### B. UNSCENTED KALMAN FILTER

The UKF was proposed in [24] as a derivative-free alternative to the EKF. Whereas the EKF's linearisation process incorporates inherent flaws (i.e. the expressions are approximated using a first-order Taylor series), the UKF addresses them by utilizing a deterministic *sampling* strategy [25], where essentially, a set of points are propagated through the true nonlinearity, without approximation. i.e. the unscented transformation uses a set of appropriately chosen weighted points to parameterise the means and covariances of probability

**Input:**  $\Delta t, z_i, \text{New\_range}$   
**Output:** Next target state estimation  $\hat{\mathbf{x}}_k$   
**if** *\_Init\_* **then** Initialize:  
    |  $\mathbf{F}, \mathbf{R}, \mathbf{Q}, \mathbf{P}_0, \hat{\mathbf{x}}_0$   
**end**  
Predict step:  
 $\hat{\mathbf{x}}_k = \mathbf{F}\hat{\mathbf{x}}_{k-1}$   
 $\mathbf{P}_k = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}$   
**if** *New\_range* **then** Update step:  
    |  $\mathbf{H}_x = \mathbf{H}(\hat{\mathbf{x}}_k)$   
    |  $\mathbf{S} = \mathbf{H}_x\mathbf{P}\mathbf{H}_x^T + \mathbf{R}$   
    |  $k = \mathbf{P}\mathbf{H}_x^T\mathbf{S}^{-1}$   
    |  $h_x = h(\hat{\mathbf{x}}_k)$   
    |  $y = z_k - h_x$   
    |  $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k + ky$   
    |  $\mathbf{I}_k = \mathbf{I}_{k-1} - k\mathbf{H}_x$   
    |  $\mathbf{P}_k = \mathbf{I}\mathbf{P}_k\mathbf{I}^T + k\mathbf{R}k^T$   
**end**

**Algorithm 1:** EKF method for Range-Only and Single-Beacon target tracking.

distributions. These points, called sigma points  $\chi$ , are propagated through the system using the state transition matrix  $\mathbf{F}$  presented in (6).

Different methods can be used to choose the sigma points (e.g. [26] and [27]). Here, the method presented in [27] has been used, where  $\chi \in \mathbb{R}^{(n \times 2n+1)}$  is defined as

$$\chi = \begin{bmatrix} \hat{\mathbf{x}}^T \\ \hat{\mathbf{x}} + \gamma\sqrt{\mathbf{P}} \\ \hat{\mathbf{x}} - \gamma\sqrt{\mathbf{P}} \end{bmatrix}, \quad (10)$$

where  $\gamma = \sqrt{n + \lambda}$ , with  $\lambda = \alpha^2(n + k) - n$ . These constant values are usually set as follows:  $\alpha$  is chosen between  $1e^{-4}$  and 1, which determines the spread of the sigma points;  $k$  is set to 0 for state estimation; and  $\beta = 2$  for Gaussian distributions.

Finally, these sigma points are weighted as follows,

$$W_0^{(\mathbf{x})} = \frac{\lambda}{n + \lambda}, \quad (11)$$

$$W_0^{(\mathbf{P})} = \frac{\lambda}{n + \lambda} + 1 - \alpha^2 + \beta, \quad (12)$$

$$W_i^{(\mathbf{x})} = W_i^{(\mathbf{P})} = \frac{1}{2(n + \lambda)}, i \in \{1, \dots, 2n\}. \quad (13)$$

The equations (5)-(9) presented in the previous section are also used in the UKF. Following the notation of [24] and [27] the UKF for ROSB tracking is described in Algorithm 2.

### C. MAXIMUM A POSTERIORI ESTIMATION

The MAP estimation [17] is a well-known method for target tracking problems. Although, the MAP estimator had long been considered to be too computationally intensive for real-time applications [28], it is becoming more commonly used thanks to processor improvements (e.g. [16] and the references therein).

**Input:**  $\Delta t, z_i, \text{New\_range}$   
**Output:** Next target state estimation  $\hat{\mathbf{x}}_k$   
**if** *\_Init\_* **then** Initialize:  
    |  $\mathbf{F}, \mathbf{R}, \mathbf{Q}, \mathbf{P}_0, \hat{\mathbf{x}}_0$   
**end**  
Predict step:  
 $\chi = [\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{x}}_{k-1} + \gamma\sqrt{\mathbf{P}_{k-1}}, \hat{\mathbf{x}}_{k-1} - \gamma\sqrt{\mathbf{P}_{k-1}}]$   
 $\chi_{\mathbf{F}} = \mathbf{F}\chi$   
 $\hat{\mathbf{x}}_k = \sum_{i=0}^{2n} W_i^{(\mathbf{x})} \chi_{\mathbf{F}}$   
 $\mathbf{P}_k = \sum_{i=0}^{2n} W_i^{(\mathbf{P})} [\chi_{\mathbf{F}} - \hat{\mathbf{x}}_k][\chi_{\mathbf{F}} - \hat{\mathbf{x}}_k]^T + \mathbf{Q}$   
**if** *New\_range* **then** Update step:  
    |  $\chi = [\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_k + \gamma\sqrt{\mathbf{P}_k}, \hat{\mathbf{x}}_k - \gamma\sqrt{\mathbf{P}_k}]$   
    |  $h_{\chi} = h(\chi)$   
    |  $\hat{y}_k = \sum_{i=0}^{2n} W_i^{(\mathbf{x})} h_{\chi,i}$   
    |  $\mathbf{P}_{h_{\chi}\hat{y}} = \sum_{i=0}^{2n} W_i^{(\mathbf{P})} [h_{\chi,i} - \hat{y}_k][h_{\chi,i} - \hat{y}_k]^T + \mathbf{R}$   
    |  $\mathbf{P}_{\hat{\mathbf{x}}\hat{y}} = \sum_{i=0}^{2n} W_i^{(\mathbf{P})} [\chi_i - \hat{\mathbf{x}}_k][h_{\chi,i} - \hat{y}_k]^T$   
    |  $k = \mathbf{P}_{\hat{\mathbf{x}}\hat{y}}\mathbf{P}_{h_{\chi}\hat{y}}^{-1}$   
    |  $y = z_k - \hat{y}_k$   
    |  $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k + ky$   
    |  $\mathbf{P}_k = \mathbf{P}_k - k\mathbf{P}_{h_{\chi}\hat{y}}k^T$   
**end**

**Algorithm 2:** UKF for Range-Only and Single-Beacon target tracking.

The EKF addresses the non-linear estimation problems by applying linearization methods, which introduce inherent errors. While UKF has been developed as an alternative strategy to address these errors, it only refines the current state, being unable to refine past linearized points (see the previous Section). In contrast, the MAP estimator computes the estimations of all states at all time steps, by using all available measurements.

The main equations of the MAP estimator are described below (adapted from [17] and [16]). Firstly, as mentioned, all available information is used to estimate the entire target trajectory by stacking all states in the time interval  $[0, \dots, k]$ ,

$$\mathbf{x}_{0:k} = [\mathbf{x}_0^T \mathbf{x}_1^T \dots \mathbf{x}_k^T]^T. \quad (14)$$

Then, the entire state vector is estimated by maximising the posterior probability density function as follows

$$p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) \propto p(\mathbf{x}_0) \prod_{k=1}^k p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{z}_k | \mathbf{x}_k), \quad (15)$$

where  $p(\mathbf{x}_0) \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$  is the prior distribution. By applying Bayes' rule, and assuming a Gaussian and independent noise in both measurement and state functions, plus

using the target motion model (5) and the range measurement model (8) explained above, (15) can be rewritten as

$$\begin{aligned}
 p(\mathbf{x}_{0:k} | \mathbf{z}_{0:k}) \propto & \\
 & \underbrace{\frac{1}{\sqrt{(2\pi)^{2n} |\mathbf{P}_0|}} \exp\left(-\frac{1}{2} \|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|_{\mathbf{P}_0}^2\right)}_{\text{Initial state constraint}} \times \\
 & \underbrace{\prod_{k=1}^k \frac{1}{\sqrt{(2\pi)^{2n} |\mathbf{Q}_{k-1}|}} \exp\left(-\frac{1}{2} \|\mathbf{x}_k - \mathbf{F}\mathbf{x}_{k-1}\|_{\mathbf{Q}_{k-1}}^2\right)}_{\text{State transition constraint}} \times \\
 & \underbrace{\prod_{k=1}^k \frac{1}{\sqrt{2\pi\sigma_{wk}^2}} \exp\left(-\frac{1}{2} \|z_k - h(\mathbf{x}_k)\|_{\sigma_{wk}^2}^2\right)}_{\text{Measurement constraint}}, \tag{16}
 \end{aligned}$$

where  $\|\mathbf{a}\|_{\mathbf{M}} \triangleq \mathbf{a}^T \mathbf{M}^{-1} \mathbf{a}$ .

Using the monotonicity of the negative logarithm, the maximisation of (16) is equivalent to the minimisation of the following cost function

$$\begin{aligned}
 c(\mathbf{x}_{0:k}) = & \frac{1}{2} \|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|_{\mathbf{P}_0}^2 + \sum_{k=1}^k \frac{1}{2} \|\mathbf{x}_k - \mathbf{F}\mathbf{x}_{k-1}\|_{\mathbf{Q}_{k-1}}^2 \\
 & + \sum_{k=1}^k \frac{1}{2} \|z_k - h(\mathbf{x}_k)\|_{\sigma_{wk}^2}^2, \tag{17}
 \end{aligned}$$

Due to the non-linearity of the measurement model (8), there is no straightforward solution. A standard approach for its optimization is to employ iterative algorithms, which can find the solution from an initial estimation  $\hat{\mathbf{x}}_0$  based on the recursion of

$$\hat{\mathbf{x}}_{0:k}^{i+1} = \hat{\mathbf{x}}_{0:k}^i + \delta \mathbf{x}_{0:k}^i, \tag{18}$$

where the parameter  $\delta \mathbf{x}_{0:k}^i$  is a correction factor, which indicates the step size and its direction. Employing the Newton-Raphson iterative minimisation method [29], as in previous works (e.g. [20]), and following [16],  $\delta \mathbf{x}_{0:k}^i$  can be found solving the linear system

$$\delta \mathbf{x}_{0:k}^i = -\mathbf{A}^{i-1} \mathbf{b}^i, \tag{19}$$

where  $\mathbf{A}$  and  $\mathbf{b}$  are the Jacobian ( $\nabla$ ) and Hessian ( $\nabla^2$ ) of the cost function (17) with respect to all stacked states (14), evaluated at the latest state estimation, which can be obtained as

$$\begin{aligned}
 \mathbf{b}^i = & \mathbf{\Pi}^T \mathbf{P}_0^{-1} (\hat{\mathbf{x}}_{0:k}^i - \hat{\mathbf{x}}_0) \\
 & + \sum_{k=1}^k \mathbf{F}_{k-1}^i{}^T \mathbf{Q}_{k-1}^{-1} (\hat{\mathbf{x}}_k^i - \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1}^i), \tag{20} \\
 & + \sum_{k=1}^k \mathbf{H}_k^i{}^T \mathbf{R}_k^{-1} (z_k - h(\hat{\mathbf{x}}_k^i))
 \end{aligned}$$

and

$$\begin{aligned}
 \mathbf{A}^i \simeq & \mathbf{\Pi}^T \mathbf{P}_0^{-1} \mathbf{\Pi} \\
 & + \sum_{k=1}^k \mathbf{F}_{k-1}^i{}^T \mathbf{Q}_{k-1}^{-1} \mathbf{F}_{k-1}^i, \tag{21} \\
 & + \sum_{k=1}^k \mathbf{H}_k^i{}^T \mathbf{R}_k^{-1} \mathbf{H}_k^i
 \end{aligned}$$

where  $\mathbf{\Pi}$ ,  $\mathbf{H}$ , and  $\mathbf{F}$  are used to adjust the dimension of a single state estimation to the entire stacked state, which have the following structure

$$\mathbf{\Pi} = [\mathbf{I}_{2n} \mathbf{0} \cdots \mathbf{0}], \tag{22}$$

$$\mathbf{H} = [\mathbf{0}_{1 \times 2n} \cdots -\mathbf{H} \cdots \mathbf{0}_{1 \times 2n}], \tag{23}$$

$$\mathbf{F} = [\mathbf{0}_{2n \times 2n} \cdots -\mathbf{F} \mathbf{I}_{2n} \cdots \mathbf{0}_{2n \times 2n}]. \tag{24}$$

Therefore, the MAP algorithm can be formulated using these equations as shown in Algorithm 3.

**Input:**  $\Delta t, z_i, \text{New\_range}$

**Output:** Next target state estimation  $\hat{\mathbf{x}}_k$

**if** *\_Init\_* **then** Initialize:

    |  $\mathbf{F}, \mathbf{R}, \mathbf{Q}, \mathbf{P}_0, \hat{\mathbf{x}}_0$

**end**

Predict step:

$\hat{\mathbf{x}}_k = \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1}$

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

**if** *New\_range* **then** Update step:

$\mathbf{x}_{0:k} = [\mathbf{x}_0^T \mathbf{x}_1^T \cdots \mathbf{x}_k^T]^T$

    Refine all the states using Newton-Raphson iterative minimisation algorithm:

**while**  $i < \max$  **or**  $\nabla c(\mathbf{x}_{0:k}) > \min$  **do**

        | Find  $\mathbf{b}^i$  and  $\mathbf{A}^i$  using (20) and (21)

        |  $\delta \mathbf{x}_{0:k}^i = -\mathbf{A}^{i-1} \mathbf{b}^i$

        |  $\hat{\mathbf{x}}_{0:k}^{i+1} = \hat{\mathbf{x}}_{0:k}^i + \delta \mathbf{x}_{0:k}^i$ ,

**end**

**end**

**Algorithm 3:** MAP for Range-Only and Single-Beacon target tracking.

Finally, a marginalization method can be used to reduce the computational cost of stacking all the states, which at a certain point can be computationally intractable. Different marginalization methods have been developed to discard old states, which are not affected significantly by a new measurement available at the current target position. For example, in [16] the Schur complement is used. However, as will be shown, a simple sliding window can also be applied with good results, where at each time-step  $k$  the state vector is updated with a new state, while the oldest one is discarded. As a result, the stacked state vector always has the same size, and therefore, the computational cost does not change.



#### D. PARTICLE FILTER

Despite the benefits of the above algorithms, the EKF, the UKF or the MAP all have difficulties in tracking multi-modal probability density functions, which is a usual problem in ROSB tracking methods [30]. Only a few estimators are specifically designed to treat multi-modal distributions. Nowadays, the PF is one of the most commonly used [31] and [32].

The PF solves, in a non-parametric way, the probability distribution problem of the HMM using the Bayes' rule (1) with the recursion of

$$p(\mathbf{x}_k | \mathbf{z}_{:k-1}) = \sum_{\mathbf{x}_{k-1}} \underbrace{p(\mathbf{x}_k | \mathbf{x}_{k-1})}_{\text{Motion model}} \underbrace{p(\mathbf{x}_{k-1} | \mathbf{z}_{:k-1})}_{\text{Particles}}, \quad (25)$$

and

$$p(\mathbf{x}_k | \mathbf{z}_{:k}) \propto \underbrace{p(\mathbf{z}_k | \mathbf{x}_k)}_{\text{Importance weights}} \underbrace{p(\mathbf{x}_k | \mathbf{z}_{:k-1})}_{\text{Particles}}, \quad (26)$$

where a bunch of particles  $\mathbf{x} \in \mathbb{R}^{2n}$  are spread on a 2D area, and are used to represent different possible states. Equation (25) represents the prediction step, which uses the motion model presented in (5) to move each particle with some random noise. In this case, the mean of all these particles represents the prior probability distribution. Then, using (26), each particle is weighted with a likelihood ratio based on the measurement probability function

$$W_k^n = \frac{1}{\sqrt{2\pi\sigma_{wk}^2}} \exp\left(-\frac{(h(\mathbf{x}_k^n) - z_k)^2}{2\sigma_{wk}^2}\right), \quad (27)$$

which calculates the probability of the state  $\mathbf{x}_k^n$  for one dimensional Gaussian function with mean equal to the distance between the observer and the particle  $h(\mathbf{x}_k^n)$ , which is the measurement model described in (8), and variance equal to  $\sigma_{wk}^2$ . In this case, the index  $n \in \{0, \dots, N\}$  indicates the particle number up to  $N$ .

Finally, all the particles are resampled according to their weight in order to obtain the posterior probability distribution and to estimate the target's position. Different resampling methods have been developed [33], where the Systematic method offers a good performance in terms of computational complexity and resampling quality. However, [34], demonstrated that other methods, such as the Compound strategy, have better performance under fast target manoeuvre circumstances.

The Compound method consists of a twofold strategy: a standard Systematic resampling method for  $(N - \ell)$  particles; and a Random resampling method for the last  $(\ell)$  particles, which are dropped randomly inside a circular area around the previous target position that has been estimated as  $\hat{\mathbf{x}}_{k-1}$ . This strategy maintains particles near the target in all directions, improving the PF's time response in front of unexpected target position variations. Moreover, it maintains the particles' spatial variability, which helps to reduce the common degeneracy problem in the PF.

Using all these considerations, Algorithm 4 can be used to track underwater targets using the PF.

**Input:**  $\Delta t, z_i, \text{New\_range}$

**Output:** Next target state estimation  $\hat{\mathbf{x}}_k$

**if** *\_Init\_* **then** Initialize:

$\mathbf{F}, \mathbf{Q}, \hat{\mathbf{x}}_0$

    The state vector for each particle and its weight associated are also initialised:

$\{\mathbf{x}_0^n\}_{n=1}^N \sim p(\mathbf{x}_0)$

$\{W_0^n\}_{n=1}^N = 1/N_p$

**end**

Predict step (25):

$\{\hat{\mathbf{x}}_k^n\}_{n=1}^N = \mathbf{F}_{k-1} \{\hat{\mathbf{x}}_{k-1}^n\}_{n=1}^N + \mathbf{Q}_{k-1}$

**if** *New\_range* **then** update step (26):

    Importance weight update using (27)

$\{W_k^n\}_{n=1}^N$

    Normalize the importance weights

$\{W_k^n\}_{n=1}^N = \{W_k^n\}_{n=1}^N / \sum_{j=1}^N W_k^j$

    Resampling:

$c = [W_k^0, W_k^{i-1} + W_k^i, \dots, W_k^{N-1} + W_k^N]$  for

$i = \{1, \dots, N - 1\}$

$u = \text{random}() / (N - \ell)$

$i = 0$

**for**  $j$  **in**  $\text{range}(N - \ell)$  **do**

**while**  $u > c^i$  **do**

$i += 1$

**end**

$\mathbf{aux}^j = \mathbf{x}_k^i$

$u += 1 / (N - \ell)$

**end**

**for**  $i$  **in**  $\text{range}(\ell)$  **do**

$\mathbf{aux}^{j+i+1} = \text{random}(\mathbf{x})$

**end**

$\{\mathbf{x}_k^n\}_{n=1}^N = \mathbf{aux}$

$\hat{\mathbf{x}}_k = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_k^n W_k^n$

**end**

**Algorithm 4:** PF for Range-Only and Single-Beacon target tracking.

### III. OPTIMAL PATH

One of the first problems to solve in underwater target tracking is to determine the path that should be followed by the observer in order to increase the accuracy of the target estimations. The ROSB methods suffers from the multi-modal distribution estimation, which is difficult to solve using standard algorithms such as the EKF. Different solutions have been found to solve this problem. For example, EKF or MAP filters have been used in parallel, where each filter tracks one possible trajectory, and a cost function is derived to find the most probable target path. One example of this is the RP-EKF [35], where each EKF uses a different initial range estimation to track a target using bearings-only, and the bank-MAP in [16], where the authors used different MAP estimators for Range-Only target tracking.

Other authors (e.g. [13], [6], [36], and [37]) have solved the multimodal problem from the system's observability perspective: i.e. by driving specific paths to maximise the amount of information or quality of the measurements conducted, affecting the accuracy of the estimated target position. Using an optimum path, the multimodal problem can be avoided, and consequently, the tracking algorithms can compute the correct target position. These studies, which are based on the Fisher Information Matrix [38], determined that a circular trajectory centred over the target maximises the system's observability. In [6], the optimal circumference for target tracking using a surface vehicle was derived not only analytically, but with field tests. Whereas the ideal circumference is one with a radius as large as possible, in real scenarios the maximum radius is typically a few hundred metres due to time constraints. For example, for a typical velocity of 1 m/s, an boat will need more than 50 min to conducts one trajectory with 500 m of radius, and moreover, the power consumption and battery limitations should also be taken into consideration. This circumference can be written as follows

$$\mathbf{p}_k^* = \mathbf{q}_k + d_k^* \mathbf{g}(\beta_k), \quad (28)$$

where  $\mathbf{p}_k^*$  is the optimal position of the tracker,  $\mathbf{q}_k$  is the target's position,  $\mathbf{g}(\beta_k) = [\cos\beta_k \ \sin\beta_k]^T$  with  $\beta_k = 2m^{-1}\pi k$ , where  $m$  is the number of range measurements per circle, and  $d_k^*$  is the optimal circumference's radius.

On the other hand, in [7] the optimal radius was estimated when the tracker's velocity was taken into account according to the larger magnitude of

$$d_k = | -\|\mathbf{q}_k - \mathbf{p}_{k-1}^*\| \cos\theta_k \pm \sqrt{\alpha_k^2 - (\|\mathbf{q}_k - \mathbf{p}_{k-1}^*\| \sin\theta_k)} |, \quad (29)$$

where  $\alpha_k$  is the tracker's displacement at time  $k$ , defined by the time elapsed  $\Delta T$  and the tracker's velocity  $\bar{v}$  as  $\alpha_k = \Delta T \bar{v}_k$ , and  $\theta_k$  is the angle between the vectors  $(\mathbf{q}_k - \mathbf{p}_{k-1}^*)$  and  $\mathbf{g}(\beta_k)$ .

The maximum radius in (29) has two main boundaries: the tracker's maximum velocity  $\bar{v}$ ; and the number of range measurements  $m$  to be conducted for each circumference, both implicitly defined in  $\alpha_k$  and  $\theta_k$ . As the tracker's velocity is constrained by the specifications of the vehicle being used, the only parameter that can be adjusted is the number of measurements. If a small number of measurements is used, a small circular radius will be obtained, otherwise the number of measurements must be increased. However, another option is to introduce additional time steps  $M \in \{1, \dots, M\}$  between measurements to increase the circumference's radius while respecting the vehicle's maximum velocity. In this case the circumference is defined by  $\beta_k = 2(Mm)^{-1}\pi k$ .

Therefore, considering all these factors, a circular tracker path with a constant radius of 100 m ( $\sim 10$  min per circle) and range measurements every 40 s have been used in our study, in order to evaluate the performance of each ROSB method.

#### IV. SIMULATIONS

A set of simulations have been conducted for the following scenarios: (a) Static Target Tracking; (b) Mobile Target Tracking; and (c) Multi-Target Tracking. These simulations have been performed in order to characterize the performance of each method described above, and tested them under different noise levels.

In general, Gaussian noise with zero mean and unit variance during the process of measurement is typically assumed [19] in analytical developments, whereas it has been observed that in some cases this is not accurate [6]. A Gaussian noise with non-zero mean and non-unit variance errors,  $w \sim \mathcal{N}(\epsilon_w, \sigma_w^2)$ , introduces a systematic error and a random uncertainty respectively in field measurements. Moreover, some measurements can differ substantially from the true range (i.e. outliers), with a potentially strong influence on the estimations. During simulations, different outlier measurements were randomly introduced by multiplying the real range by four. The total number of outlier measurements did not exceed the 1% of the total number of measurements. Therefore, each scenario has been studied using Monte Carlo Simulation (MCS) methods [39], where different noise levels have been added to each range measurement to evaluate the tracking method's robustness in the face of them, and to obtain a more realistic simulation.

To characterise the tracking filters' performance, the step response criteria has been used. The step response concept is generally used in control system analysis to characterise the time evolution of a dynamic system [40, Fig. 5-2]. It is known that the system's response has two components: transient response and steady state response. The transient response is present in the short period of time immediately after the system is turned on or a change is conducted on the input control. If the system is asymptotically stable, then the transient disappears and the system is determined by its steady state component only. Under this assumption, the ROSB methods presented in this paper can be characterised by

- Settling Time ( $T_S$ ): Time required to reach and stay below a threshold error
- Recovery Time ( $T_R$ ): Time required to reach and stay below a threshold error after a step response
- Steady State Error ( $\epsilon_{SS}$ ): The error between the real target position and its estimation in the limit as time goes to infinity
- Root Mean Square Error (RMSE): The X-Y error between the true target position and its estimation computed as  $\text{RMSE} = \sqrt{\mathbb{E}\{\|\mathbf{x}_k - \hat{\mathbf{x}}_k\|^2\}}$

Finally, with all these considerations, the main parameters used to conduct the simulations are described below:

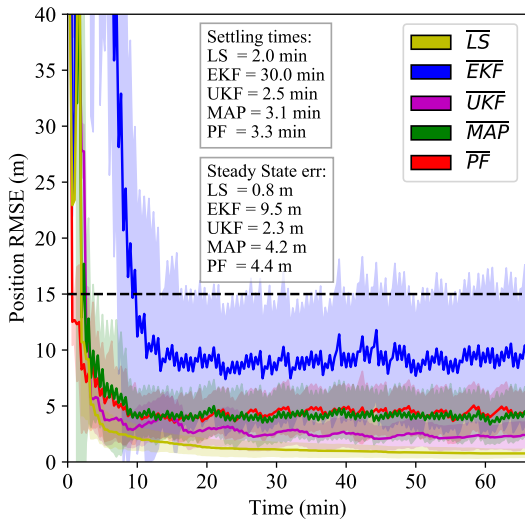
- Observer velocity: 1 m/s
- Observer circumference radius: 100 m
- Target velocity: 0 or 0.2 m/s (variable among tests)
- Random range noise: 1 or 4 m (variable among tests)
- Systematic range noise: up to 1%

- Range measurement outliers: up to 1%
- Time between iterations: 20 s
- Time between range measurements: 40 s
- Simulation steps: 200
- MCS iterations: 100
- Number of particles (PF): 3000

**A. STATIC TARGET LOCALISATION**

The target localisation algorithms explained in the previous section were first tested in a static scenario. This scenario is used as an initial test to evaluate the performance of the ROSB methods. Moreover, we were able to compare them against standard target localisation methods such as the LS [6], which is a good estimator to localise targets in static scenarios.

Fig. 1 shows the simulated RMSE average value and its Standard Deviation (STD) after 100 MCS iterations, showing the filters’ time response. This test was conducted with a range noise equal to  $\epsilon_w = 1\%$  and  $\sigma_w = 4$  m. The fastest algorithm to reach an RMSE lower than 15 m is the LS (2.0 min), which is also the algorithm with the lowest  $\epsilon_{SS}$  (0.8 m). On the other hand, the EKF provided the worst performance with an  $\epsilon_{SS}$  of 9.5 m, and a  $T_S$  of 30 min, i.e. it did not estimate the target’s position with very high accuracy.

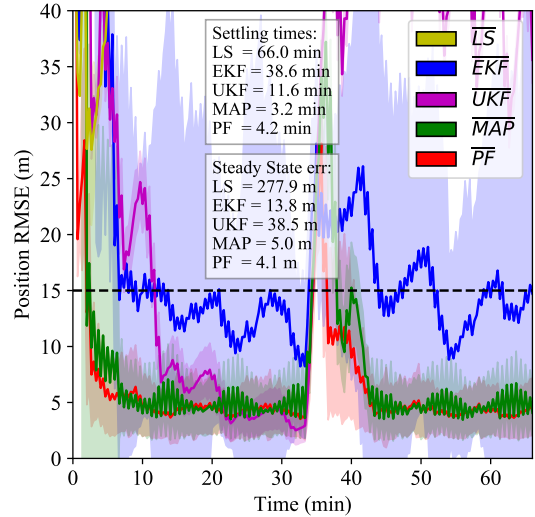


**FIGURE 1.** Algorithms’ time response for static target localisation. The settling time and the steady state error have been derived for the EKF, UKF, PF, MAP and LS algorithms. Results obtained through 100 MCS iterations with  $\epsilon_w = 1\%$  and  $\sigma_w = 4$  m. The dark colours represent the average RMSE whereas the light colours represent their standard deviation.

Table 2 shows the simulated filters’ performance under other range noise parameters. The LS method is the best one in many scenarios. However, the most robust filter in the face of outliers is the PF, which has an average  $T_S = 11$  min and  $\epsilon_{SS} = 8.8$  m. In contrast, the inclusion of a systematic error was not relevant to the filters’ performance. It is known that concentric circumferences around a target constitute the most robust path in the face of systematic errors, as explained in [6].

**B. MOBILE TARGET TRACKING**

In this section we discuss the results of simulations, where a mobile target with constant velocity of 0.2 m/s plus a 90° right turn after the 100<sup>th</sup> step (i.e. at 35 min from starting) was used as the second testing scenario. Besides the  $T_S$  and the  $\epsilon_{SS}$ , the  $T_R$  was also computed as the filter’s time response after the right turn, when the accuracy of the target’s trajectory is lost, see Fig. 2.



**FIGURE 2.** Algorithms’ time response for dynamic target tracking. The settling time and the steady state error are derived for the EKF, UKF, PF, MAP and LS algorithms. Moreover, the increase of the RMSE due to the target’s right turn can also be observed. Results obtained through 100 MCS iterations with  $\epsilon_w = 1\%$  and  $\sigma_w = 4$  m. The dark colours represent the average RMSE whereas the light colours represent their standard deviation.

The result of these indicators ( $T_S$ ,  $T_R$ , and  $\epsilon_{SS}$ ) after 100 MCS and for different configurations of range noise are shown in Table 3. In general, the PF algorithm out-performed all the other methods, followed by the MAP algorithm. On the other hand, the LS was unable to track the mobile target as expected. The same performance is observed in all the noise cases, and for all the indicators studied.

We now discuss the performance of the PF using different configurations, e.g. the contribution of different resampling methods, the number of particles used, and the computational time required.

1) PF: Resampling Method

The Compound (C) method has the advantage of sudden response in case of fast changes in the target’s direction (see Section II-D), by combining Systematic and Random resampling algorithms. Therefore, it can be adjusted by modifying the ratio between the number of particles used for the Systematic method ( $N_{sys}$ ) and for the Random method ( $N_{rand}$ ) respectively.

We conducted a set of simulations using different resampling methods as: (1) Multimodal, (2) Systematic, and (3) Compound. The results obtained in the time domain are presented in Table 4. After different iterations, we observed



**TABLE 2.** Filters' performance for different range measurement errors: Static scenario.

Parameter	Filter	Mean (STD) <sup>a</sup>	Mean (STD) <sup>b</sup>	Mean (STD) <sup>c</sup>	Mean (STD) <sup>d</sup>
$T_S$ (min)	LS	1.7 (0.0)	<b>2.0 (0.5)</b>	<b>2.0 (0.5)</b>	26.5 (31.0)
	EKF	9.0 (0.4)	33.5 (16.7)	30.0 (14.5)	38.5 (19.6)
	UKF	2.3 (0.0)	2.4 (0.2)	2.5 (0.3)	15.8 (21.3)
	MAP	1.7 (0.3)	2.8 (2.4)	3.1 (3.0)	16.7 (22.9)
	PF	<b>0.3 (0.4)</b>	2.3 (3.5)	3.3 (7.5)	<b>11.0 (15.5)</b>
$\varepsilon_{SS}$ (m)	LS	<b>0.2 (0.1)</b>	<b>0.7 (0.4)</b>	<b>0.8 (0.4)</b>	13.0 (2.6)
	EKF	2.1 (1.3)	10.0 (6.3)	9.5 (6.2)	36.2 (74.2)
	UKF	0.6 (0.3)	2.3 (1.2)	2.3 (1.2)	<b>8.8 (15.2)</b>
	MAP	1.0 (0.5)	4.0 (2.1)	4.2 (2.2)	25.6 (49.5)
	PF	3.1 (1.4)	4.2 (2.1)	4.4 (2.0)	8.8 (17.0)

The range error introduced at each range measurement was as follows:

<sup>a</sup> ( $\sigma_w^2 = 1\text{m}$ )

<sup>b</sup> ( $\sigma_w^2 = 4\text{m}$ )

<sup>c</sup> ( $\sigma_w^2 = 4\text{m}$ ,  $\epsilon_w = 1\%$ )

<sup>d</sup> ( $\sigma_w^2 = 4\text{m}$ ,  $\epsilon_w = 1\%$ , plus 1% of outliers)

**TABLE 3.** Filters' performance for different range measurement errors: Dynamic scenario.

Parameter	Filter	Mean (STD) <sup>a</sup>	Mean (STD) <sup>b</sup>	Mean (STD) <sup>c</sup>	Mean (STD) <sup>d</sup>
$T_S$ (min)	LS	66.0 (0.0)	66.0 (0.0)	66.0 (0.0)	66.0 (0.0)
	EKF	6.2 (0.3)	45.8 (19.4)	38.6 (19.0)	46.9 (20.1)
	UKF	11.7 (0.1)	11.5 (0.4)	11.6 (0.4)	26.1 (22.3)
	MAP	1.9 (0.5)	5.1 (9.6)	<b>3.2 (2.7)</b>	21.2 (24.5)
	PF	<b>1.7 (0.3)</b>	<b>4.0 (7.4)</b>	4.2 (5.0)	<b>17.0 (20.8)</b>
$T_R$ (min)	LS	30.0 (0.0)	30.0 (0.0)	30.0 (0.0)	30.0 (0.0)
	EKF	8.1 (1.3)	28.1 (4.0)	28.6 (3.4)	29.7 (4.6)
	UKF	34.6 (0.1)	34.5 (0.2)	34.5 (0.2)	18.7 (25.2)
	MAP	6.0 (0.1)	7.8 (3.6)	<b>8.8 (3.8)</b>	19.8 (25.6)
	PF	<b>5.8 (1.5)</b>	<b>7.4 (3.2)</b>	8.8 (5.1)	<b>15.1 (25.3)</b>
$\varepsilon_{SS}$ (m)	LS	272.6 (0.1)	272.6 (0.3)	277.8 (0.2)	405.4 (2.5)
	EKF	4.3 (1.5)	12.7 (18.8)	13.8 (6.3)	37.0 (94.2)
	UKF	38.7 (0.4)	38.6 (1.8)	38.5 (1.7)	<b>8.0 (15.9)</b>
	MAP	2.5 (0.6)	4.8 (2.3)	5.0 (2.3)	16.7 (38.9)
	PF	<b>1.0 (0.5)</b>	<b>3.8 (2.1)</b>	<b>4.1 (2.2)</b>	10.3 (22.8)

The range error introduced at each range measurement was as follows:

<sup>a</sup> ( $\sigma_w^2 = 1\text{m}$ )

<sup>b</sup> ( $\sigma_w^2 = 4\text{m}$ )

<sup>c</sup> ( $\sigma_w^2 = 4\text{m}$ ,  $\epsilon_w = 1\%$ )

<sup>d</sup> ( $\sigma_w^2 = 4\text{m}$ ,  $\epsilon_w = 1\%$ , plus 1% of outliers)

that the best ratio ( $r(\%) = N_{rand}/N_{sys} * 100$ ) for the Compound resampling method was  $\sim 6.7$  (i.e. only 6.7% of particles are resampled using the Random method).

## 2) PF: Number of Particles

The number of particles also has an important impact on the PF's performance. The more particles used to represent the target's position, the more accurate its estimation will be. In Table 5 the results for 1000, 3000, 6000, and 10000 particles are shown. In all these simulations, the Compound method with ratio 6.7 was used.

## 3) Processing Time Required

Finally, the processing time can be an important constraint and a decisive factor to choose one or another method. While processing time may not be a limiting factor in some underwater scenarios due to slow dynamic processes involved, in some cases this may not be true. For example, in centralised multi-target tracking situations, the total time required to compute all the targets' positions could increase significantly with the number of targets, and therefore, the processing time must be taken into consideration.

The algorithms' runtime performance is shown in Table 6, where clearly the PF is the most expensive method from the

**TABLE 4.** PF's performance for different resampling strategies: Dynamic scenario.

Resampling <sup>a</sup>	$T_R$ (min)	$T_S$ (min)	$\epsilon_{SS}$ (m)
	Mean (STD) <sup>b</sup>	Mean (STD) <sup>b</sup>	Mean (STD) <sup>b</sup>
Multimodal	24.7 (5.0)	3.7 (1.9)	5.7 (9.8)
Systematic	21.2 (4.6)	4.4 (2.9)	2.6 (2.9)
C( $r = 0.3$ )	13.3 (8.2)	4.4 (2.7)	3.1 (6.2)
C( $r = 1.7$ )	9.6 (5.9)	3.3 (1.2)	1.3 (0.6)
C( $r = 3.3$ )	5.8 (2.5)	<b>3.2 (0.4)</b>	1.3 (0.7)
C( $r = 6.7$ )	<b>5.6 (2.2)</b>	3.3 (0.5)	<b>1.2 (0.7)</b>
C( $r = 10.0$ )	5.9 (1.9)	3.5 (0.9)	1.4 (0.6)
C( $r = 16.7$ )	5.6 (2.9)	3.4 (1.0)	1.3 (0.7)
C( $r = 33.3$ )	12.6 (8.2)	5.2 (4.2)	2.8 (0.9)
C( $r = 66.7$ )	29 (0.0)	29 (0.0)	14.5 (2.2)

<sup>a</sup> see [33] and [34] for a detailed description of these methods

<sup>b</sup> ( $\sigma_w^2 = 1\text{m}$ )

**TABLE 5.** PF's performance for different number of particles: Dynamic scenario.

# particles	$T_R$ (min)	$T_S$ (min)	$\epsilon_{SS}$ (m)
	Mean (STD) <sup>a</sup>	Mean (STD) <sup>a</sup>	Mean (STD) <sup>a</sup>
1000	13.2 (3.9)	6.1 (3.4)	1.4 (0.7)
3000	9.6 (5.9)	3.3 (1.2)	1.3 (0.6)
6000	7.3 (3.1)	2.8 (0.7)	1.1 (0.5)
10000	<b>6.1 (2.4)</b>	<b>2.5 (0.4)</b>	<b>1.1 (0.5)</b>

<sup>a</sup> ( $\sigma_w^2 = 1\text{m}$ )

computational time point of view, whereas the MAP algorithm appears as good compromise between performance and computation time. Nonetheless, both methods are suitable for this application due to the slow dynamics in most underwater scenarios.

**TABLE 6.** Filter's average runtime required at each step.

Filter	Runtime <sup>a</sup> (ms)
EKF	0.2
LS	0.4
UKF	0.6
MAP	10.6
PF (1000)	34.4
PF (3000)	98.4
PF (6000)	181.9
PF (10000)	276.7

<sup>a</sup> Processor Intel(R) Core(TM) i7-4760HQ CPU @ 2.10 GHz with 8 GB of RAM memory

### C. MULTI-TARGET TRACKING

Multi-target Monte Carlo simulations using both static and dynamic target were used to characterise the filters' performance, in order to determine: (1) the filter's response when the tracker is not conducting its manoeuvres directly over the

target, but with some offset; and (2) the feasibility of tracking multiple targets simultaneously.

Firstly, 49 targets were spread on a grid of 7x7, each one separated 100 m from its immediate neighbour. Then, a tracker conducted circular manoeuvres over the centre of the grid with a radius of 100 m and a velocity of 1 m/s. Every 40 seconds a new range measurement was computed between the tracker and each of the 49 targets, updating an individual filter for each target in order to estimate its position. This procedure was repeated 5 times, one for each tracking algorithm.

#### 1) Multiple Static Targets

The filter's performance in a multi-static target localisation scenario is represented through coloured maps (Fig. 3). Those maps indicate the RMSE between the true target position (black triangles) and its estimation. This test was conducted 50 times using MCS iterations with a range noise equal to  $\epsilon_w = 1\%$  and  $\sigma_w = 4$  m. The average value among all RMSE is presented in Fig. 3. This style of X-Y representation is commonly used [20], which indicates the target's estimation error obtained as a function of its position with respect to the tracker's path centre. For example, the best performance achievable using the LS algorithm for static target localisation was presented in [6, Fig. 4], where the CRB was used, and then verified through real field tests. Here, not only the LS but also the MAP and PF algorithms followed a similar performance.

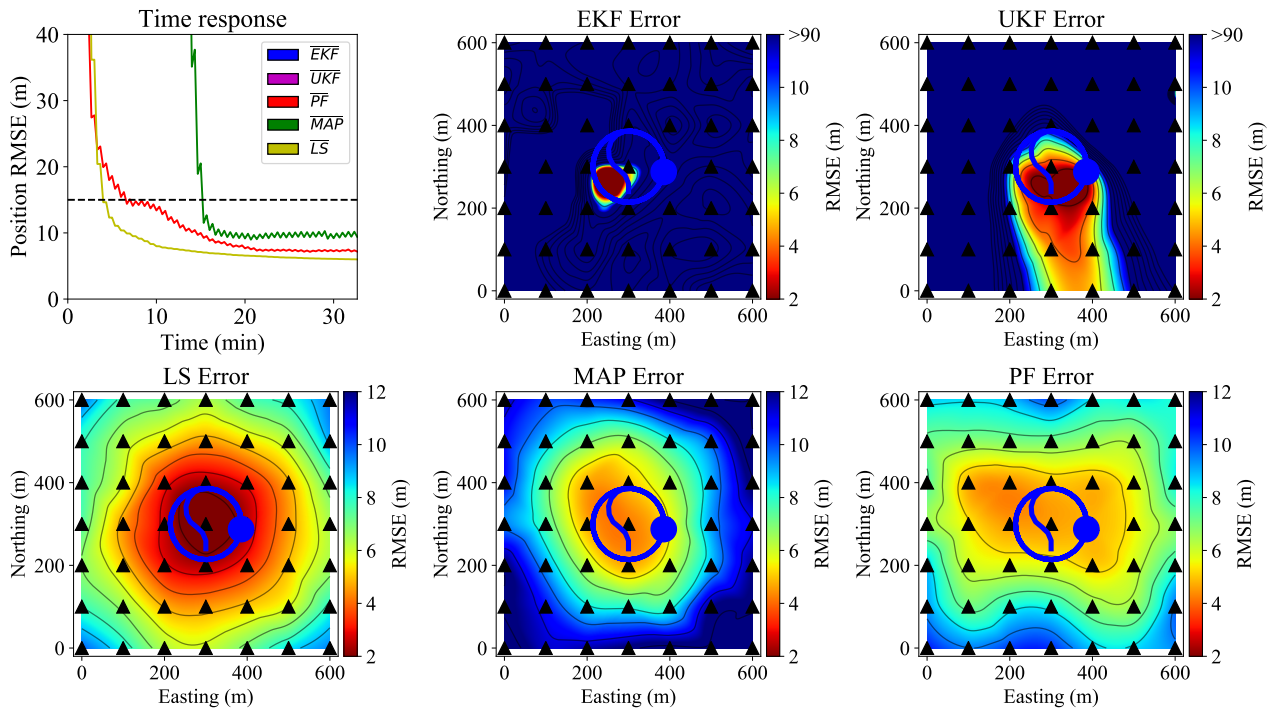
The simulations show that the LS, MAP, and PF have a superior performance, where the targets close to the centre were better estimated. This behaviour is due to the observability of the system (see Section III). On the other hand, both the EKF and UKF have the poorest performance, with more accurate estimated positions close to the centre of the tracker's path. This behaviour can be explained by the state's initialization, where the first tracker position is used as an initial target estimation. Therefore, the targets which are close to the tracker have the best initial estimation.

Finally, the average values of the 49 target estimations for the  $T_S$  and the  $\epsilon_{SS}$  are shown in Fig. 3 (top left) and in Table 7. The LS algorithm exhibited the best performance with a  $T_S = 4.2$  min and a  $\epsilon_{SS} = 6$  m. This result is similar to the result obtained in the previous section. However, here we also showed what could be expected in the case of tracking only one target which is not directly below the tracker's path centre.

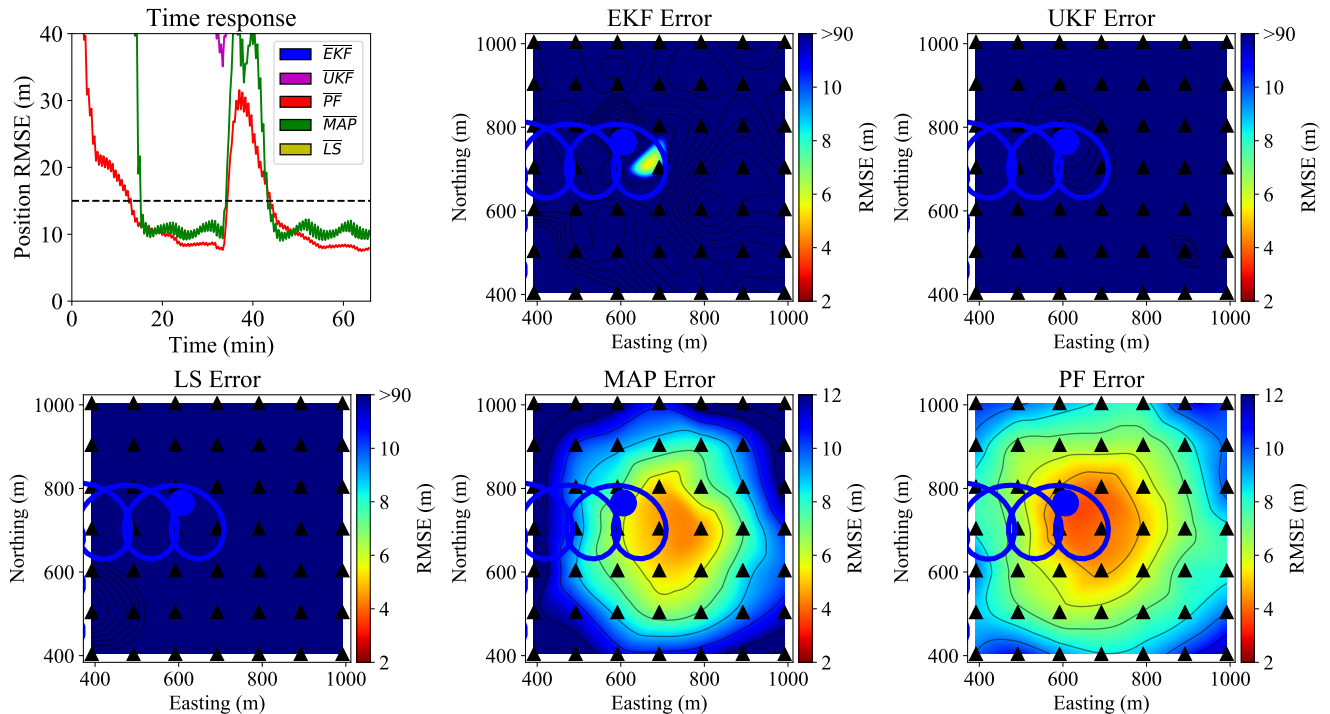
#### 2) Multiple Mobile Targets

The performance of each tracking method using mobile targets is presented in Fig. 4. The EKF and the UKF had difficulties to track the targets when these were not directly below the centre of the tracker's path. Obviously, the LS algorithm cannot accurately estimate their position either.

In Table 8, the values of  $T_R$ ,  $T_S$ , and  $\epsilon_{SS}$  are shown, where a Gaussian noise of 4 m with 1% of systematic error has been added in all range measurements. The results show that only



**FIGURE 3.** Algorithms' performance for static multi-target tracking scenario. A Gaussian error of 4 m plus a systematic error of 1% at each range measurement have been added. Each triangle represents the targets' position. The blue line and the circle represents the tracker's trajectory and its last position respectively. The colour map indicates the interpolation of the RMSE between the real target position and its estimation (the average of the last value over all the MCS iteration). The top-left figure indicates the average of the time response over all the targets' RMSE, where the methods not depicted had an error greater than 40 m, and the dashed line represents the threshold used to compute the  $T_S$ .



**FIGURE 4.** Algorithms' performance for dynamic multi-target tracking scenario. A Gaussian error of 4 m plus a systematic error of 1% at each range measurement have been added. Each triangle represents the targets' last position. The blue line and the circle represent the tracker's trajectory and its last position respectively. The colour map indicates the interpolation of the RMSE obtained at each target (the average of the last value over all the MCS iterations). The top-left figure indicates the average of the time response over all the targets' RMSE, where the methods not depicted had an error greater than 40 m, and the dashed line represents the threshold used to compute the  $T_S$ .

**TABLE 7.** Target tracking algorithms' performance in a multi-target scenario. Average results obtained from 49 targets: Static scenario.

Filter	$T_S(\text{min})$	$\varepsilon_{SS}(\text{m})$
	Mean (STD) <sup>a</sup>	Mean (STD) <sup>a</sup>
LS	<b>4.2 (1.3)</b>	<b>6.0 (0.1)</b>
EKF	29.3 (2.8)	64.5 (6.8)
UKF	25.1 (9.0)	55.0 (7.4)
MAP	15.4 (7.4)	9.7 (1.1)
PF	8.9 (4.0)	7.3 (0.4)

<sup>a</sup> ( $\sigma_w^2 = 4 \text{ m}$ ,  $\epsilon_w = 1 \%$ )

the PF and the MAP algorithms were able to track all the targets with an acceptable accuracy, where the PF had the best performance, also when they conduct a right turn (~30 min after the simulation's beginning).

**TABLE 8.** Target tracking algorithms' performance in a multi-target scenario. Average results obtained from 49 targets: Dynamic scenario.

Filter	$T_R(\text{min})$	$T_S(\text{min})$	$\varepsilon_{SS}(\text{m})$
	Mean (STD) <sup>a</sup>	Mean (STD) <sup>a</sup>	Mean (STD) <sup>a</sup>
LS	29.7 (0.0)	65.0 (0.0)	313.6 (9.1)
EKF	29.7 (0.0)	65.0 (0.0)	51.4 (5.6)
UKF	26.2 (5.7)	65.0 (0.0)	114.7 (9.7)
MAP	18.0 (9.0)	<b>11.3 (5.4)</b>	10.1 (1.5)
PF	<b>13.4 (5.8)</b>	12.3 (7.4)	<b>7.8 (0.6)</b>

<sup>a</sup> ( $\sigma_w^2 = 4 \text{ m}$ ,  $\epsilon_w = 1 \%$ )

## V. FIELD TESTS

After the study conducted using the MCS methods, different field tests were conducted in order to validate the results and conclusions derived. These tests were divided into two groups:

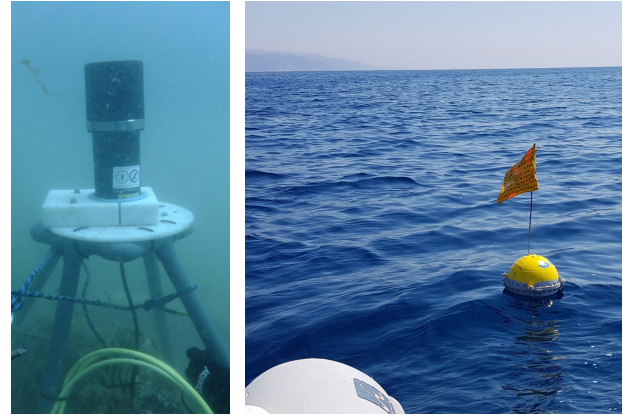
- (A) Tests carried out in the OBSEA underwater observatory ([www.obsea.es](http://www.obsea.es)) of the Universitat Politècnica de Catalunya (UPC)
  - 1) Static test
  - 2) Dynamic test
- (B) Dynamic tests conducted in Monterey Bay with the support of the Monterey Bay Aquarium Research Institute (MBARI) ([www.mbari.org](http://www.mbari.org))

During those tests, the main parameters were the same ones used in Section IV to be able to perform the appropriate comparisons (e.g. number of particles (PF) = 3000).

### A. OBSEA TESTS

These tests consisted in localizing three static targets and tracking a mobile one located at the coastal cabled observatory OBSEA. In both cases, S2C-18/34 acoustic modems from the EvoLogics company were used to measure ranges between observer and targets. These modems use the Sweep-Spread Carrier (S2C) technology, delivering an excellent performance, and working at 18 - 34 kHz. For these tests, a

small boat was employed as observer, which computes the target position using the different methods studied. Fig. 5 shows one of the modems deployed in the seafloor used in the static test (left picture), and the drifter buoy used in the dynamic test (right picture).



**FIGURE 5.** Underwater photography of the USBL (left) used as a target to localise during the static test, and the drifter buoy (right) used as a target to track during the dynamic test conducted in the OBSEA observatory.

#### 1) Static test at OBSEA

The first experiment carried out at OBSEA was designed to localise three acoustic modems previously deployed. One was attached to the observatory's buoy (M3) at 5 m depth, and two other modems deployed on the seafloor (M1 and M2), near the observatory's junction box at 20 m depth. Moreover, one of the seafloor modems had USBL capabilities (M1).

The slant range between the boat and each modem is represented in Fig. 6, where only one outlier (out of more than 300 measurements) between the boat and the USBL was obtained. However, this outlier must be taken into consideration, which has an important implication in the performance of the localisation algorithms, especially for the LS method as observed in Table 2. Therefore, it has to be removed in order to obtain an accurate estimation, whereas the PF is more robust even with the outlier included.

Fig. 7 shows a geographic Cartesian coordinate map, where the target position estimations using the LS, MAP and PF, and the boat path conducted are represented. The PF algorithm has been executed 10 times in order to observe the prediction's variability due to the inherent random processes involved. The estimation of the second modem (M2) using the MAP algorithm has an important error, which can be caused by the lack of observability in the system. The M2 was the farther modem from the centre of the boat path, and as observed in Section IV-C, not only the steady state error is worse, but also the settling time, which is important in such situations. Under these circumstances the MAP algorithm needs more time to obtain an accurate estimation of the target's position.



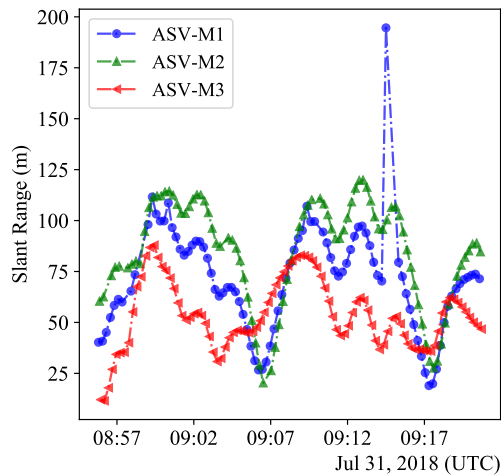


FIGURE 6. Slant ranges obtained between the boat and the underwater modems deployed at OBSEA.

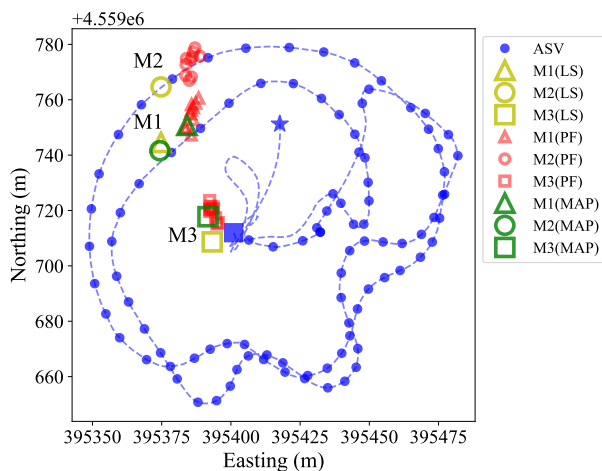


FIGURE 7. Static target localisation results obtained in the OBSEA observatory. The small blue dots represent the X-Y coordinates where a range measurement between the boat and each underwater target were carried out. The dotted blue line represents the boat's trajectory, where the start and square dots are the start and end positions respectively. Finally, the estimated positions for M1, M2 and M3 modems using LS, PF, and MAP algorithms are also represented.

Finally, the results obtained during this experiment can be compared against the results obtained through simulations as depicted in Table 9. Unfortunately, the true deployment position was not available, and therefore, the RMSE could not be computed. In this case, the error between the slant range measured with the modems and the theoretical slant range computed using the targets' prediction was used, denoted as Real  $\varepsilon_{SS}$ (Mean). On the other hand, the STD variation of the latest 10 estimations of each filter were used to obtain an indicator of its variability, denoted as Real  $\varepsilon_{SS}$ (STD). Therefore, the values are not equal, however, the error's proportion is the same. On the other hand, both the EKF and the UKF were not taken into consideration during the field tests because of their lower performance, especially for

TABLE 9. Target tracking algorithms' performance for multi-target localisation purposes. Simulations vs real field tests: Static scenario. Test conducted on July 8, 2018.

# Filter	Sim $\varepsilon_{SS}$ (m)		Real $\varepsilon_{SS}$ (m)
	Mean (STD) <sup>a</sup>	Mean (STD) <sup>b</sup>	Mean (STD)
LS	0.8 (0.4)	6.0 (0.1)	8.2 (0.0)
EKF	9.5 (6.2)	64.5 (6.8)	– (–)
UKF	2.3 (1.2)	55.0 (7.4)	– (–)
MAP	4.2 (2.2)	9.7 (1.1)	29.3 (5.9)
PF	4.4 (2.0)	7.3 (0.4)	10.7 (3.3)

<sup>a</sup> ( $\sigma_w^2 = 4$  m,  $\epsilon_w = 1$  %) from static scenario (Table 2)

<sup>b</sup> ( $\sigma_w^2 = 4$  m,  $\epsilon_w = 1$  %) from static multi-target scenario (Table 7)

targets not centred below the tracker's path (Table 7).

## 2) Dynamic test at OBSEA

The second experiment carried out in the OBSEA was designed to track a dynamic target, which was a drifting buoy with an acoustic modem. The results obtained are presented in Fig. 8, where the boat path (blue dotted line), the range measurements (blue dots), the real target position (black dotted line), the PF estimation (red dots), and the MAP estimation (green dots) are represented. On the other hand, the inset graphic (in the bottom-right corner) shows the RMSE between the estimated target position and its real position. Whereas the communication with the drifter was lost around 10:10 h UTC, the boat was able to track the drifter as soon as the communication was available again.

However, it has to be taken into consideration that the real target position was interpolated using its initial deployment and recovery positions, and the sea currents present during the test, where the GPS position was not available during this experiment. Therefore, the mean error computed during this test should not be taken strictly into consideration, but as an indicator of the filter's performance. For example, as demonstrated in simulations, the MAP algorithm has a Recovery Time greater than the PF algorithm, and such performance is also observed in this field test, see Fig. 9.

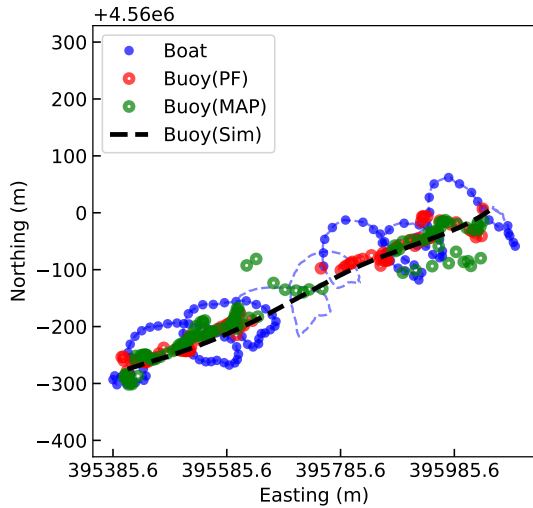
## B. MBARI TESTS

Finally, a last test in Monterey Bay California in collaboration with the Monterey Aquarium Research Institute (MBARI) was conducted in order to validate the algorithms and observe their performance under real conditions. This test was performed using a Wave Glider (WG) from the Liquid Robotics company, which tracked a Coastal Profiling Float (CPF) [41] for more than 15 hours, Fig. 10.

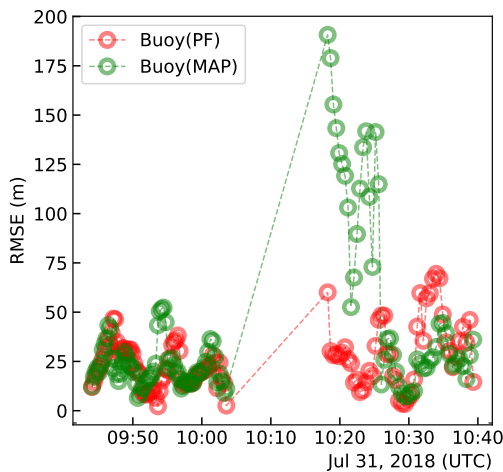
The CPF is a device which spends the majority of its time static, resting on the seabed. It periodically goes to the surface to fix a GPS position, then it drifts with the sea currents a few metres until it conducts another immersion to return to the seabed. During the test, the CPF conducted three immersions to  $\sim 60$  m depth, as depicted in Fig. 11 (red line).

In order to know the CPF's position, the WG conducted circular paths around the area while periodically measuring





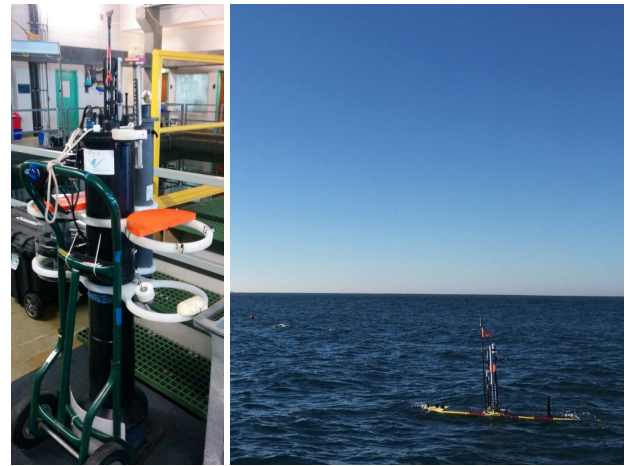
**FIGURE 8.** Dynamic target tracking results obtained in the OBSEA observatory. The small blue dots represents the X-Y coordinates where a range measurement between the boat and drifter was carried out. The dotted blue line represents the boat trajectory. The red dots represent the target's estimation using the PF algorithm, whereas the green dots represent the target's estimation using the MAP algorithm. The black dashed line is the drifter trajectory.



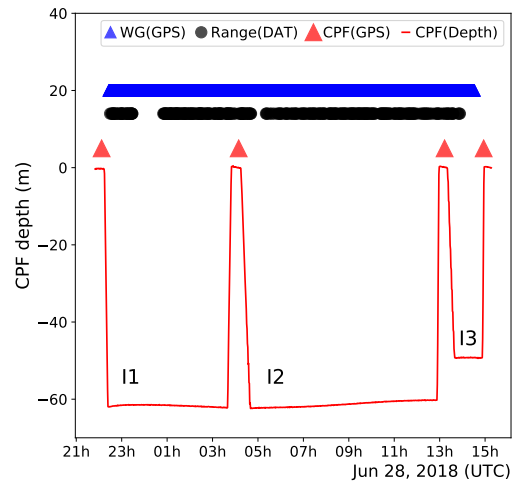
**FIGURE 9.** Dynamic target tracking results obtained in the OBSEA observatory. This graphic represents the RMSE between the real and the estimated target's position using both the PF and the MAP estimation methods.

the slant range to the CPF. An acoustic modem (ATM-900) from the Teledyne Benthos company was installed in the CPF and a Benthos DAT modem installed in the WG to measure the ranges. Both devices work at 16-21 kHz, and use a phase shift keying modulation technique. The Benthos DAT modem is a standard acoustic modem which also has USBL capabilities. Because of that, a comparative study between the target's position obtained with the USBL and the position estimated using the ROSB tracking algorithms explained above could be performed.

Fig. 12 shows the path conducted by the Wave Glider and the CPF, both obtained using their own GPS. Moreover, the



**FIGURE 10.** Coastal Profile Float (left) and Wave Glider (right) during sea tests conducted at Monterey Bay, California.

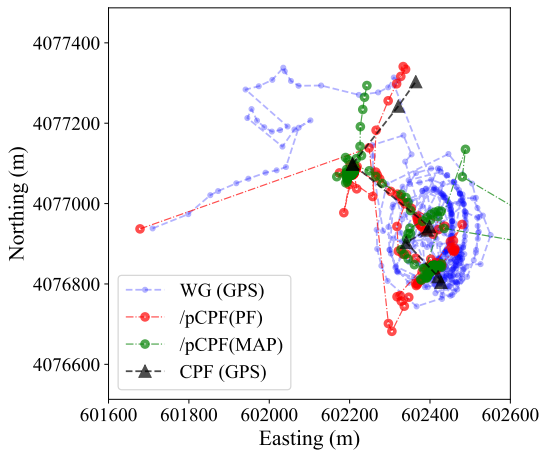


**FIGURE 11.** The CPF's depth profile (red line), when a CPF (red triangles) and a Wave Glider (blue triangles) positions were fixed, and when a slant range measurement was conducted between both devices (black dots) are presented in this graph.

CPF estimated path computed using the PF (red dots) and the MAP (green dots) algorithms are also shown.

In addition, Fig. 13 shows the RMSE between the real CPF position and its estimations provided by the USBL system, and the PF and the MAP algorithms. The real CPF position has been computed by using the GPS positions while on the surface. However, it should be taken into consideration that no "true" CPF position while on the seabed was available. Therefore, this can cause an increase in the average error. For example, the CPF's displacement produced by sea currents during the immersion has not been taken into consideration to compute the real CPF position.

On the other hand, the error obtained from USBL is much greater than the error obtained from the PF or MAP algorithms. In general, a USBL system has to be calibrated in advance, especially to eliminate the attitude misalignment between the acoustic transducer and the Inertial Measurement



**FIGURE 12.** The Wave Glider trajectory during the field test WG(GPS), the CPF path CPF(GPS), and the CPF estimated position using PF /pCPF(PF) and MAP estimation /pCPF(MAP), are shown in this figure.

**TABLE 10.** Target tracking algorithm’s performance for target tracking. Simulations vs real field tests: Dynamic scenario. Test conducted on July 31, 2018.

# Filter	Sim $\epsilon_{SS}$ (m)		Real $\epsilon_{SS}$ (m)
	Mean (STD) <sup>a</sup>	Mean (STD) <sup>b</sup>	Mean (STD)
LS	277.8 (0.2)	313.6 (9.1)	– (–)
EKF	9.3 (6.3)	51.4 (5.6)	– (–)
UKF	38.3 (1.7)	114.7 (9.7)	– (–)
MAP	4.8 (2.3)	10.1 (1.5)	22.4 (2.3)
PF	4.0 (2.2)	7.8 (0.6)	21.9 (0.8)

<sup>a</sup> ( $\sigma_w^2 = 4$  m,  $\epsilon_w = 1$  %) from dynamic scenario (Table 3)

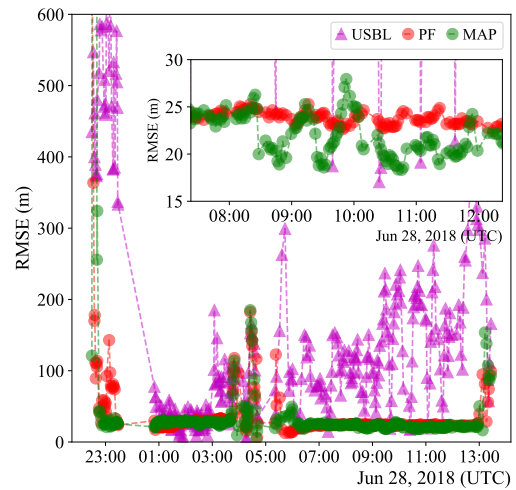
<sup>b</sup> ( $\sigma_w^2 = 4$  m,  $\epsilon_w = 1$  %) from dynamic multi-target scenario (Table 8)

Unit (IMU), and also to adjust their internal clocks. However, despite the calibrations, large errors can be expected due to the sea state when a USBL instrument is installed on small platforms, such as a Wave Glider. Nonetheless, the error measured during this test is something unexpected, and therefore, indicates a poor calibration or some undetected misalignment.

Finally, the inset of Fig. 13 shows a zoom of PF’ and MAP’ RMSE results, where a mean of 21.9 m (PF) and 22.4 m (MAP) with an STD of 0.8 (PF) and 2.3 (MAP) have been obtained, which can be compared to the results obtained through the simulations as shown in Table 10.

## VI. DISCUSSION

The aim of this paper is the study and development of different algorithms to track targets with autonomous marine vehicles moving along an horizontal plane by using range-only methods, such approach reduces the cost, power requirements, and complexity over other methods (e.g. using a USBL system which also requires an IMU). The methods presented in this paper may improve autonomous target tracking as a key factor for maritime and industries activities. For example, in the framework of fishery management (i.e.



**FIGURE 13.** RMSE between the real CPF position and its estimation. Comparison between USBL (purple triangles), PF algorithm (red dots), and MAP algorithm (green dots). Inset shows a close look of the RMSE between 8:00 and 12:00 hours.

producing ancillary data for fishery management of relevant commercial items, as the Norway lobster (*Nephrops norvegicus*), as well as snow crabs [42]), where multiple platforms intercommunication protocols and autonomous navigation capabilities should be developed (i.e. the acoustic tracking of emitters placed on freely moving animals, spreading out from repopulating marine reserves).

Our data may also be useful in developing autonomous networks to monitor and quantify human impacts, as described by the Marine Strategy Framework Directive of the European Commission [43]. The spatial scaling of data gathered at fixed observatories, could be complemented by the use of flexible and adaptive networks of monitors and autonomous underwater vehicles. Our data could help toward the implementation of multi-parametric coordinated monitoring.

In this work, an extended study has been carried out focusing on performance comparisons among different ROSB algorithms (LS, EKF, UKF, MAP, and PF), specifically designed for 3 typical underwater scenarios: localising a static target, tracking a dynamic target, and multi-target tracking. The MCS method provides a close comparison between the simulations and real field tests conducted. Simulations are powerful tools which allow a close study of more complex and noisy/real scenarios, compared to strictly deterministic analytical studies.

For example, Table 9 summarises the field test results conducted in the cabled observatory station OBSEA ([www.obsea.es](http://www.obsea.es)) to localise multiple targets, which have been compared with the Steady State Error ( $\epsilon_{SS}$ ) among 100 MCS for single target localization (with 4 m of Gaussian error and 1% of systematic error), and 100 MCS for multi-target localisation (with 4 m of Gaussian error and 1% of systematic error). The results show that the target which is in the centre of the tracker path is estimated better. Finally, the real results

show a greater error than the simulations, however, the error's proportion is the same, where the LS is the best algorithms whereas the MAP is the worst method. One can observe that neither the EKF nor UKF algorithms have been considered for real field tests. Moreover, the variability of these filters (STD) also follows the same trend presented in the simulations. It should also be noted that the real position of each modem was not available, and the mean error presented has been obtained using the slant range measured with the acoustic modems, and the slant range computed using their estimated positions. Furthermore, the greatest error presented in the field tests may be due to the lower number of ranges used, since the more measurements used, the greater the accuracy can be achieved.

The dynamic target tracking test conducted in the OBSEA, where a drifter buoy was used as a target, and presented in Fig. 8, shows the performance of PF and MAP algorithms in real field tests. The inset shows the evolution of the RMSE over time, where one can clearly observe that the PF has a settling time faster than that of the MAP. This behaviour has been observed previously using MCS (e.g. Fig. 4).

Finally, the test conducted in Monterey Bay with the PF and the MAP algorithms can be compared as before with simulations, see Table 10. The field test performance is shown to be quite similar to that of the simulations. However, in this case one should take into consideration that the real position of the CPF was only obtained when it was on the surface, using a GPS. Therefore, its displacement while it conducted the immersion and before it settled down on the seafloor could not be computed. This means that the RMSE presented in Fig. 13 (inset) may have an inherent error, although the general performance was demonstrated.

## VII. CONCLUSION

This work shows the performance of different algorithms under different scenarios with the objective of tracking underwater target by using autonomous vehicles. The main mathematical notation of each algorithm, and their performance under simulations and field tests have been conducted, and the best practice has been derived. From a methodological point of view, this work advances the understanding of accuracy that can be achieved by using ROSB target tracking methods with autonomous vehicles.

The algorithms considered in this study are LS, EKF, UKF, MAP, and PF. All these algorithms have been compared with each other. Simulations and experimental results suggest that an accuracy of a few metres can be achieved using the PF, which we demonstrated to be the fastest and the most accurate algorithm with respect to other studied approaches to estimate an underwater target position especially when this target is moving. For example, in a simulated dynamic scenario with a quasi ideal noise measurement of 1m, the PF achieves a settling time equal to 1.7 min, a recovery time equal to 5.8 min, and a steady state error of 1 m, but it also has more accurate values than the other algorithms in noisier cases.

## ACKNOWLEDGMENT

We gratefully acknowledge the assistance and support of Brent Jones and other MBARI engineers, and the David and Lucile Packard foundation. This work has been lead and carried out by members of the Tecnoterra associated unit of the Scientific Research Council through the Universitat Politècnica de Catalunya, the Jaume Almera Earth Sciences Institute and the Marine Science Institute (ICM-CSIC).

## REFERENCES

- [1] W. S. Burdick, *Underwater Acoustic System Analysis*. US: Peninsula Publishing, 2003.
- [2] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870505000168>
- [3] M. M. Hunt, W. M. Marquet, D. A. Moller, K. R. Peal, W. K. Smith, and R. C. Spindel, "An acoustic navigation system," *Technical Report WHOI-74-6*, vol. 74, no. 6, pp. 1–65, 1974.
- [4] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: recent advances and new challenges," vol. 88, 2006.
- [5] L. Paull, S. Saeedi, M. Seto, and H. Li, "Auv navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, Jan 2014.
- [6] I. Masmija, S. Gomariz, J. Del-Rio, B. Kieft, T. O'Reilly, P.-J. Bouvet, and J. Aguzzi, "Optimal path shape for range-only underwater target localization using a wave glider," *The International Journal of Robotics Research*, vol. 37, no. 12, pp. 1447–1462, 2018.
- [7] N. Crasta, D. Moreno-Salinas, A. Pascoal, and J. Aranda, "Multiple autonomous surface vehicle motion planning for cooperative range-based underwater target localization," *Annual Reviews in Control*, vol. 46, pp. 326–342, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S136757881830107X>
- [8] J. Kalwa, D. Tietjen, M. Carreiro-Silva, J. Fontes, L. Brignone, N. Gracias, P. Rídao, M. Pflingstorn, A. Birk, T. Glotzbach, S. Eckstein, M. Caccia, J. Alves, T. Furfaro, J. Ribeiro, and A. Pascoal, "The european project morph: Distributed uuv systems for multimodal, 3d underwater surveys," *Marine Technology Society Journal*, vol. 50, no. 4, pp. 26–41, 2016.
- [9] R. Kaune, J. HÄ́urst, and W. Koch, "Accuracy analysis for tdoa localization in sensor networks," in *14th International Conference on Information Fusion*, July 2011, pp. 1–8.
- [10] B. Zhang, Y. Hu, H. Wang, and Z. Zhuang, "Underwater source localization using tdoa and fdoa measurements with unknown propagation speed and sensor parameter errors," *IEEE Access*, vol. 6, pp. 36 645–36 661, 2018.
- [11] C. Zheng, D. Sun, L. Cai, and X. Li, "Mobile node localization in underwater wireless networks," *IEEE Access*, vol. 6, pp. 17 232–17 244, 2018.
- [12] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: a survey," *Telecommunication Systems*, vol. 52, no. 4, pp. 2419–2436, Apr 2013.
- [13] D. Moreno-Salinas, A. Pascoal, and J. Aranda, "Optimal sensor placement for acoustic underwater target positioning with range-only measurements," *IEEE Journal of Oceanic Engineering*, vol. 41, no. 3, pp. 620–643, July 2016.
- [14] G. Vallicrosa, P. Rídao, D. Ribas, and A. Palomer, "Active range-only beacon localization for auv homing," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 2286–2291.
- [15] N. Crasta, D. Moreno-Salinas, B. Bayat, A. M. Pascoal, and J. Aranda, "Range-based underwater target localization using an autonomous surface vehicle: Observability analysis," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, April 2018, pp. 487–496.
- [16] G. Huang, K. Zhou, N. Trawny, and S. I. Roumeliotis, "A bank of maximum a posteriori (map) estimators for target tracking," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 85–103, Feb 2015.
- [17] J. Gauvain and C. H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, April 1994.
- [18] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*. Athena Scientific; Edition: 2nd, 2008.



- [19] M. Wüthrich, S. Trimpe, C. G. Cifuentes, D. Kappler, and S. Schaal, "A new perspective and extension of the gaussian filter," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1731–1749, 2016.
- [20] A. Alcocer, "Positioning and navigation systems for robotic underwater vehicles," PhD dissertation, Universidade Técnica de Lisboa Instituto superior Técnico, 2009.
- [21] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2nd edition, Belmont, MA, 1995.
- [22] O. Cappe, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential monte carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, May 2007.
- [23] A. H. Jazwinski, *Stochastic processes and filtering theory*. Analytical Mechanics Associates, Inc. Academic Press., 1970.
- [24] New extension of the Kalman filter to nonlinear systems, vol. 3068, 1997. [Online]. Available: <https://doi.org/10.1117/12.280797>
- [25] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, March 2004.
- [26] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, March 2000.
- [27] R. V. der Merwe and E. A. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221), vol. 6, May 2001, pp. 3461–3464 vol.6.
- [28] Batch maximum likelihood (ML) and maximum a posteriori (MAP) estimation with process noise for tracking applications, vol. 5204, 2004. [Online]. Available: <https://doi.org/10.1117/12.506442>
- [29] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment — a modern synthesis," in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 298–372.
- [30] G. P. M. Rozál and J. A. Hartigan, "The map test for multimodality," *Journal of Classification*, vol. 11, no. 1, pp. 5–36, Mar 1994. [Online]. Available: <https://doi.org/10.1007/BF01201021>
- [31] F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, and F. Hlawatsch, "Distributed localization and tracking of mobile networks including noncooperative objects," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 1, pp. 57–71, March 2016.
- [32] C. Forney, E. Manii, M. Farris, M. A. Moline, C. G. Lowe, and C. M. Clark, "Tracking of a tagged leopard shark with an auv: Sensor calibration and state estimation," in 2012 IEEE International Conference on Robotics and Automation, May 2012, pp. 5315–5321.
- [33] T. Li, M. Bolic, and P. M. Djuric, "Resampling methods for particle filtering: Classification, implementation, and strategies," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 70–86, May 2015.
- [34] I. Masmijtja, S. Gomariz, J. Del-Río, P.-J. Bouvet, and J. Aguzzi, "Underwater multi-target tracking with particle filters," in *OCEANS 2018 MTS/IEEE Kobe*, May 2018, pp. 1–5.
- [35] P. D. Pradeep, D. A. Kumar, and G. Sahu, "Estimating the target tracking using a set of range-parameters for gain modification using kalman filters," *International Journal of Microwave Engineering (JMICO)*, vol. 1, no. 1, January 2016.
- [36] B. T. Hinson, M. K. Binder, and K. A. Morgansen, "Path planning to optimize observability in a planar uniform flow field," in 2013 American Control Conference, June 2013, pp. 1392–1399.
- [37] J. D. Quenzer and K. A. Morgansen, "Observability based control in range-only underwater vehicle localization," in 2014 American Control Conference, June 2014, pp. 4702–4707.
- [38] H. L. V. Trees, K. L. Bell, and Z. Tian, *Detection Estimation and Modulation Theory, Part I: Detection, Estimation, and Filtering Theory*, 2nd Edition. John Wiley & Sons, Inc., 2013.
- [39] D. W. Heermann, *Computer Simulation Methods in Theoretical Physics*. Springer-Verlag, Berlin, Germany, 1990.
- [40] B. C. K. Farid Golnaraghi, *Automatic Control Systems* 9th. John Wiley and Sons Ltd; Edition: 9th Revised edition, 2009.
- [41] B. Ha, "Coastal profiling float depth control," MBARI Intern Report, 2018.
- [42] P. Elliman, "Snow crab tracking using wave gliders," Hydro international, 2017.
- [43] "MSFD (Marine Strategy Framework Directive) 2008. Directive 2008/56/EC of the European Parliament and of the Council of 17 June 2008 establishing a framework for community action in the field of marine environmental policy."



institutions: the MBARI, the Institut Supérieur de l'Électronique et du Numérique (ISEN), and the Institute of Marine Sciences (ICM-CSIC).



SPARTACUS GOMARIZ received the bachelor's, M.Sc., and Ph.D. degrees in telecommunication engineering from the UPC, Barcelona, Spain, in 1990, 1995, and 2003, respectively. He is currently an Associate Professor for the Department of Electronic Engineering, the UPC, and a member of the research group remote acquisition systems and data processing (SARTI). From 1990 to 2009, he was a Teaching Assistant for the Vilanova i la Geltrú School of Engineering, and from 2010 to 2015 for the Escola Universitària d'Enginyeria Tècnica Industrial de Barcelona. In 2016, he joined the staff of the Barcelona East School of Engineering, UPC. His research interests include linear and nonlinear control theory, gain scheduled control, fuzzy control, design of navigation, guidance, and control systems for underwater vehicles. He is currently a member of the following societies: the IEEE Oceanic Engineering Society and the Spanish Committee of Automation of the International Federation of Automatic Control. Moreover, he is involved in Spanish Research Networks: Marine Robotics and Automation and Instrumentation and Applied Technology for the Study, Characterization, and Sustainable Exploration of Marine Environment (MarInTech).



his research on Electronic Instrumentation, Interoperability in Marine Sensor Networks and Wireless Sensor Networks



integration for ongoing collaborative field programs and engineering tests.

IVAN MASMITJA MS in electronic engineering at the UPC. Currently, he is conducting his PhD in underwater target tracking using acoustic-range-only methods and moving platforms at the UPC. His field of expertise is underwater technology, involved in both underwater cabled observatories and autonomous vehicles. He has been working at SARTI-MAR research group for the UPC since 2011, participating in different national and international projects, and collaborating with different

SPARTACUS GOMARIZ received the bachelor's, M.Sc., and Ph.D. degrees in telecommunication engineering from the UPC, Barcelona, Spain, in 1990, 1995, and 2003, respectively. He is currently an Associate Professor for the Department of Electronic Engineering, the UPC, and a member of the research group remote acquisition systems and data processing (SARTI). From 1990 to 2009, he was a Teaching Assistant for the Vilanova i la Geltrú School of Engineering, and from 2010 to 2015 for the Escola Universitària d'Enginyeria Tècnica Industrial de Barcelona. In 2016, he joined the staff of the Barcelona East School of Engineering, UPC. His research interests include linear and nonlinear control theory, gain scheduled control, fuzzy control, design of navigation, guidance, and control systems for underwater vehicles. He is currently a member of the following societies: the IEEE Oceanic Engineering Society and the Spanish Committee of Automation of the International Federation of Automatic Control. Moreover, he is involved in Spanish Research Networks: Marine Robotics and Automation and Instrumentation and Applied Technology for the Study, Characterization, and Sustainable Exploration of Marine Environment (MarInTech).

JOAQUIN DEL-RIO was born in Catalonia, Spain, in 1976. He received his Bs and Ms degrees in Telecommunication Engineering and Electronic Engineering, in 1999 and 2002, respectively, both from the Technical University of Catalonia, Spain. Since 2001, he is Assistant Professor in the Electronic Engineering Department of the Universitat Politècnica de Catalunya, Barcelona, Spain. Member of the research group "Remote acquisition systems and data processing (SARTI)", focusing his research on Electronic Instrumentation, Interoperability in Marine Sensor Networks and Wireless Sensor Networks

BRIAN KIEFT has been an embedded software engineer at MBARI, in Monterey Bay, California, since 2006. He has worked on various platforms, including mooring controllers, benthic instruments, ASVs, and several AUVs and their associated payloads. Brian currently works on development of the Tethys AUV—a long-range, upper-water-column AUV designed primarily for biological sensing. Apart from development, Brian also takes part in mission planning and payload integration for ongoing collaborative field programs and engineering tests.



**TOM O'REILLY** is a software engineer at MBARI. His interests include software and applications for autonomous underwater vehicles (AUVs) and autonomous surface vehicles, including navigation and tracking, adaptive sampling, and development of small low-power cytometers and microscopes for use by microbial oceanographers. He is also interested in standard protocols for marine sensor interfaces and networks, and helped to standardize the OGC PUCK interface originally developed at MBARI. Tom has been working with science colleagues to use Android devices for "crowd-sourcing" and data acquisition applications.



**PIERRE-JEAN BOUVET (M'10)** was born in Clermont-Ferrand, France, in 1978. He received the dipl.-ing. and Ph. D degrees in electrical engineering from the National Institute of Applied Science (INSA) Rennes, France, in 2001 and 2005, respectively. In 2005, he joined NXP semiconductors (formerly Philips) in Caen, France, as baseband decoding architect for digital TV demodulator products (DVB-T/T2). Since 2009, he has been an associate professor in the department of embedded systems, acoustic and communications at the engineering college of ISEN, Brest, France. His current research interests include underwater acoustic communications, underwater network, MIMO transmission, iterative reception, synchronization and localisation algorithms.



**JACOPO AGUZZI** is a marine ecologist working with fisheries that is to date broadly interested in the monitoring of marine environments and the status of the services they provide by autonomous robotic technologies. He dedicated his research to the understanding of how biodiversity and ecosystem functioning are affected by behavioural rhythms of megafauna, which bias our sampling outcomes due to massive populational movements into the benthopelagic 3D scenario. Since the past decade, he is conducting an extensive monitoring activity on species behavioural rhythms and resulting community dynamism by cabled seafloor video-observatories and their docked mobile platforms, from the coastal areas to the deep-sea. As a result of that, he progressively got more interested in robotic technological solutions allowing the establishment of autonomous monitoring networks of cabled observatories and mobile platforms as well as in automated video-imaging for biological data extraction (i.e. species tracking and classification) and storage in multidisciplinary data banks.

...