Received: 26 March 2018 | Revised: 18 December 2018 | Accepted: 31 January 2019

DOI: 10.1002/asjc.2107

SPECIAL ISSUE

WILEY

A data-driven particle filter for terrain based navigation of sensor-limited autonomous underwater vehicles

José Melo[1] | Aníbal Matos[1,2]

[1]INESC TEC - INESC Technology and Science, Porto, Portugal

[2]FEUP - Faculty of Engineering, University of Porto, Porto, Portugal

Correspondence
José Melo, INESC TEC - INESC Technology and Science, Porto, Portugal.
Email: jose.melo@fe.up.pt

Abstract

In this article a new Data-Driven formulation of the Particle Filter framework is proposed. The new formulation is able to learn an approximate proposal distribution from previous data. By doing so, the need to explicitly model all the disturbances that might affect the system is relaxed. Such characteristics are particularly suited for Terrain Based Navigation for sensor-limited AUVs, where typical scenarios often include non-negligible sources of noise affecting the system, which are unknown and hard to model. Numerical results are presented that demonstrate the superior accuracy, robustness and efficiency of the proposed Data-Driven approach.

KEYWORDS

autonomous underwater vehicles, data-driven learning, particle filter, terrain based navigation, underwater navigation

1 | INTRODUCTION

Terrain Based Navigation (TBN) is a term used to refer to a class of algorithms that takes advantage of variations of the terrain to obtain navigation position fixes, in a process that is similar to what happens for instance with the use of Global Navigation Satellite Systems (GNSS). In fact, information about the terrain, or bottom topography, can be very powerful not only for the case of TBN, but also, for example, for proximity navigation relative to drifting iceberg, as suggested in [1].

Underwater TBN is a fairly recent topic, with the first body of work on the topic dating from the early 1990s, with the initial approaches focused on using dense sensors, able to map large areas of terrain within a single measurement acquisition step. The experimental validation of such approaches was also consistently coupled with the use of high-grade INS. However, recently, the study of TBN for sensor-limited Autonomous Underwater Vehicles (AUVs) has been reported by several authors, for example [2,3]. Sensor-limited systems refer to a class of vehicles equipped with low-information sonar like Doppler Velocity Log (DVL) or altimeters, but also low accuracy inertial measurement units (IMU).

The use of low grade IMUs motivates a tightly-coupled integration between all the sensors, but also requires an online estimation of critical sensor errors 4. Accurate modelling of such errors will definitely yield better performance in terms of navigation accuracy of the system. However this requires a thorough understanding of the underlying physical properties of the system, but also a detailed modelling of those sources of error, which is not always easy or even possible to achieve. For a complete and up to date review of state of the art TBN algorithms for AUVs, the reader is referred to [5].

This article presents a novel data-driven approach to underwater TBN for sensor-limited systems. Data-driven methods do not depend on an explicit and detailed model of the environment. Instead, these methods are based on statistical models or machine learning techniques, which

try to capture trends from previously collected data. The learned trends can then be used to predict future states of the system. In this article new data-driven formulations of the Particle Filter (PF) are proposed, that can be more robust and efficient than traditional PF formulations when in presence of strong non-modelled disturbances. Moreover, this can be achieved without an explicit modelling of the drifting error sources that affect the system. A typical application foreseen for the proposed approach would be of low grade AUV, without a DVL, and navigating under the influence of currents.

The remainder of this article is organized as follows. Section 2 presents a brief overview of related work, while Section 3 introduces the Data-Driven Particle Filter. Additionally, Section 4 details on the learning process of the proposed approach. Section 5 gives some numerical results attesting the performance of the proposed approach, using two different methods, namely using Least Squares and Gaussian Processes, with Section 6 presenting some concluding remarks.

2 | BACKGROUND AND RELATED WORK

A state-space model for the underwater TBN problem can be expressed by the following difference equations:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + w_k \tag{1a}$$

$$\mathbf{z}_k = \mathcal{M}(\mathbf{x}_k, \mathbf{u}_k) - d_k + v_k \tag{1b}$$

Equation 1a represents the state transition equation. The state vector, $\mathbf{x}_k$, assumed to be Markovian, consists of the vehicle's two-dimensional horizontal position, referenced to a north-east-down earth-fixed frame. For sensor-limited systems, an augmented state vector is sometimes used, in order to accommodate not only the position, but also the vehicle attitude and angular rates, as well as critical sensor errors that might need to be estimated. $\mathbf{u}_k$ contains the position updates as calculated from the INS, and w_k represents the process noise.

The measurement model Equation 1b compares the measurements from the observation vector, $\mathbf{z}_k$, to the bathymetric map function, $\mathcal{M}$, with a projection-based scheme used to project the measured ranges into the three-dimensional space [4]. Because the vehicle is not navigating at the surface when acquiring the measurements, the depth of the vehicle, d_k, is also taken into account. Analogously to the motion model, measurement noise is described by v_k.

TBN is a strong non-linear problem, mostly due to the strong nonlinearities inherent of the terrain map $\mathcal{M}(\mathbf{x}_k, \mathbf{u}_k)$. Therefore, the interest on using non-parametric non-linear Bayesian methods, like the Particle Filter (PF) or the Point-Mass Filter (PMF), to address this problem is obvious. While both the PF and the PMF have been successfully demonstrated to handle the problem of terrain navigation, there has been a strong preference towards the use of PFs, which in fact have become the primary choice for addressing the problem of underwater TBN. Multiple authors have adopted the Sequential Importance Resampling (SIR) PF to address underwater TBN, particularly for sensor-limited systems [2,6–8]. However, it is known that in some situations the SIR-PF can fail, for example if new measurements appear at the tail of the prior distribution, or if the likelihood of the measurements is relatively too peaked. One of the root causes for this is the assumption that the process model can be used as a suitable proposal distribution of the PF. However, this is not always true, and the topic of generating better proposal densities has in fact received significant attention by researchers in the past.

The Hybrid SIR PF [9] first, and later the Unscented Particle Filter (UPF) [10] where among the first algorithms designed to generate better proposal densities. It has been theoretically demonstrated that particle filters with a proposal distribution obtained using the UKF outperform existing filters, however the additional computation requirements needed are very significant. An alternative for generating better proposal distributions, but with more modest computational requirements, has been proposed [11]. Such an approach uses a Kalman linear smoothing estimator for generating the proposal distribution. The main difference to the previous, is that only a single proposal is generated for all the particles, being in that sense a more efficient approach. However, the use of the Kalman smoother precludes its use when in presence of highly non-linear models or multimodal distributions.

In the context of TBN, the choice of a suitable proposal distribution has also been studied recently. In [7] the Mixture Particle Filter (MPF) and the Prior Particle Filter (PPF) have been proposed, both based on using a non-informative uniform distribution as proposal density. Simulation results demonstrated the superiority of the PPF in terms of the asymptotic convergence of the filters. Later, a suitable compact support for the uniform distributions was derived, using the Fisher information matrix of the terrain [12]. In a similar approach, in [13] a PF is also used, with subsets of the particles being weighted with different weighting functions.

3 | DATA-DRIVEN PARTICLE FILTERS

The Particle Filter is a numerical approximation to the recursive Bayes Filters that uses a weighted set of particle

to approximate the posterior density, $p(\mathbf{x}_k|\mathbf{z}_k)$. PFs borrow ideas from Monte Carlo methods, Importance sampling, and Resampling, in order to perform this approximation. While this representation is only approximate, as the number of sampled particles N_s increases it is guaranteed to converge for the true solution. Moreover, this approximation can represent a much broader space of distributions than, for example, the Kalman Filter, that is restricted to Gaussian distributions only [14]. Notwithstanding, this realization of the Bayes Filter suffers for the curse of dimensionality, as its complexity increases exponentially with the dimension of the problem. For a detailed insight of the PF framework the interested reader should refer to [15] and the references therein.

3.1 | SIR particle filter

The SIR-PF is, perhaps, the most widely known implementation of the particle filter framework, mostly due to its simplicity. In the SIR-PF the dynamical model $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is used as the proposal distribution $q(:)$. By sampling new particles directly from the process model, the SIR-PF overcomes the need to find an optimal proposal distribution, which is sometimes hard or even impossible to derive. The other aspect that differentiates the SIR-PF is its adaptive strategy for resampling. While other variations of the PF might resample every time step, SIR-PF adopts an adaptive strategy, using the effective number of particles, N_{eff}, for monitoring the depletion of the particle set.

The performance of the SIR-PF is then dependent of having a process model that accurately describes the system. When this is not the case, and there are non-modelled disturbances affecting the system, the process model distribution is, in some sense, distant from the true proposal distribution. In such situations, the SIR-PF is known to become very sensitive to outliers, leading to poor performance and sometimes even to divergence from the true solution. In order to alleviate this problem, the use of Robust Particle Filters has been proposed by some authors, for example [16]

3.2 | Data-driven approaches

The main idea behind the Data-Driven PF (DD-PF) here proposed is then to learn an approximate proposal density, by capturing trends from previously collected data. Such data will then be used to find a suitable proposal distribution, that is somehow better than the process model. The proposed DD-PF can be implemented following the general PF structure, according to Algorithm 1, with the only difference being the learning process, which must be performed in the beginning of each iteration, on lines 6 to 11. The nature of this learning process will be detailed in the following section.

By using previous data, the historic of previous estimated states, it is possible to predict a likely next state of the filter. This information can be valuable in generating a suitable proposal density. The proposal density, $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$, will then be approximated as

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k) \approx \mathcal{N}(\mu_D, \Sigma_D) \tag{2}$$

with $\mathcal{N}(\mu_D, \Sigma_D)$ being the Gaussian distribution with the mean and covariance matrix μ_D and Σ_D, respectively, whose values are learnt from the data using a suitable learning algorithm. For every iteration of the filter a new proposal density needs to be estimated.

The learning algorithm to be implemented will use the previous w position estimates of the filter, $\mathcal{X} = \{\tilde{\mathbf{x}}_{k-w}, \ldots, \tilde{\mathbf{x}}_{k-1}\}$, together with its respective time instants, $\mathcal{T} = \{k-w, \ldots, k-1\}$, to make an informed prediction of the new state of the filter at the current time step, $\hat{\mathbf{x}}_k$, as well as provide an estimate of the uncertainty of such prediction. These values, respectively μ_D and Σ_D, will then be used to generate an appropriate proposal density, as indicated by Equation 2.

$$\{\mu_D, \Sigma_D\} = \mathcal{L}(\mathcal{T}, \mathcal{X}) \tag{3}$$

Even though the true proposal distribution $q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$ is dependent on the observations, they are not explicitly used in the learning process. However, the contribution of previous observations is indirectly considered through the posterior of the previous time-steps. Two alternative implementations of the DD-PF were implemented, one using the Least Square algorithm for the learning process (DD-PF-LS), and the other one using Gaussian Processes (DD-PF-GP). This will be detailed in the following section. These formulations were primarily designed for addressing the problem of underwater TBN for AUVs. Therefore, this constraints the learning step to be performed online, and integrated with remaining steps of the filter.

The convergence of Particle Filters has been widely addressed in the literature, for example in [17]. In what follows some literature results that provide valuable indications about the convergence of the proposed DD-PF will be highlighted. The DD-PF differs to the standard SIR-PF only on the proposal distribution, which in the present case is approximated by a Gaussian distribution, with parameters learnt from past data.

The convergence properties of PFs with generic proposal distributions has been addressed in [18], with the authors noting that the convergence of PFs with Gaussian proposal distribution is ensured as long as the ratio of the optimal importance distribution and its approximation is bounded. Moreover, the covariance matrix should also be bounded from below. By following a similar approach, and by noting

the similarity between the KF and the proposed learning procedures, it should perhaps be possible to ensure the convergence of the proposed filters DD-PF, under certain terrain conditions. Nevertheless, and following the standard practices within the field of TBN, in the remaining sections we will present a set of Monte Carlo simulation that will be able to demonstrate the performance of the proposed filters.

Algorithm 1 Data-Driven Particle Filter Algorithm

for i= 1, ..., N **do**
  $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$
  $w_0^{(i)} = \frac{1}{N}$
end for
loop
  if k < w **then**
    $q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}) = p(\mathbf{x}^{(i)}|\mathbf{x}_{k-1}^{(i)})$
  else
    $\{\mu_D, \Sigma_D\} = \mathcal{L}(\mathcal{T}, \mathcal{X})$
    $q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}) \approx \mathcal{N}(\mu_D, \Sigma_D)$
  end if
  for i=1,...,N **do**
    $\mathbf{x}_k^{(i)} \sim q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$
  end for
  for i=1,...,N **do**
    $w_k^{(i)} = w_{k-1}^{(i)} \frac{p(\mathbf{z}_k|\mathbf{x}_k^{(i)})p(\mathbf{x}^{(i)}|\mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}$
  end for
  $t = \sum_{j=1}^{N} w_k^{(j)}$
  for i=1,...,N **do**
    $\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{t}$
  end for
  $(\mathbf{X}_k, \mathbf{W}_k) = \{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N}$
  if $N_{\text{eff}} < N_{\text{thr}}$ **then**
    $(\mathbf{X}_{k+1}, \mathbf{W}_{k+1}) \leftarrow \text{resample}(\mathbf{X}_k, \mathbf{W}_k)$
  end if
end loop

4 | LEARNING THE PROPOSAL DENSITY

For the sake of simplicity, it was chosen to decouple the learning process in its two components, x and y, respectively. Recalling the assumption that AUVs have relatively slow dynamics, characterized by smooth motions and without any sudden changes of velocity, this assumption is rather mild. Additionally, such assumption also favours a less computationally demanding learning process, as basically any correlations between the two directions are disregarded. Therefore, two independent data-driven learning mechanisms will exist. The x will be computed as

$$\{\mu_{D,x}, \Sigma_{D,xx}\} = \mathcal{L}(\mathcal{T}, \mathcal{X}_x\} \tag{4}$$

with the y component being computed in an analogous process.

To generate an appropriate Gaussian approximation of the proposal distribution the mean and covariance matrices are build such that $\mu_D = \left[\mu_{D,x}, \mu_{D,y}\right]^T$ and $\Sigma_D = diag\left(\Sigma_{D,xx}, \Sigma_{D,yy}\right)$. The obtained Gaussian distribution is going to be highly correlated with the historic data used in the learning process, particularly when using the Least Squares, which explicitly models the motion of the vehicle with constant velocity model. Nevertheless, this correlation is in fact expected, due to the assumed straight-line motion of the vehicles, and should in fact favour a good performance of the filter.

4.1 | Least squares regression

Recalling the aforementioned low-dynamics and constant velocity assumptions made above, a one-dimensional model of vehicle moving with a constant velocity can be modelled as

$$x_i = \beta_0 + \beta_1 t_i \tag{5}$$

where t_i, the input, corresponds to time, while x_i, the output, corresponds to the position of the vehicle. The parameters β_0 and β_1 correspond to the initial position of the vehicle and its velocity. By collecting a series of input and output observations, $\{t_i,\ x_i\}_{i=1}^{w}$, the parameters $\tilde{B} = [\beta_0\ \beta_1]^T$ can be determined using the closed-form expression for the Ordinary Least Squares (OLS) algorithm:

$$\tilde{B} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{Y} \tag{6}$$

where $\boldsymbol{\Phi}$ is the $w \times 2$ matrix of time inputs and $\mathbf{Y}$ is the $w \times 1$ matrix of the output position observations. In this way, it is possible to learn the parameters β that better fit previous data. Then, a new position of the vehicle, x_n, at time instant t_n, can then by extrapolated by using the estimated parameters, $\tilde{\beta}$:

$$x_n = \begin{bmatrix} 1 & t_n \end{bmatrix} \tilde{B} \tag{7}$$

This extrapolated value will the output of the data-drive learning mechanism.

The OLS algorithm just presented is adequate to estimate static parameters. However, when the parameters are time-varying, the use of a recursive version of the OLS algorithm is more appropriate. The Recursive Least Squares (RLS) algorithm is a recursive formulation of the least-square problem, where new estimates of the parameters are updated with new input and output observations. The use of a forgetting factor λ can be interpreted as a weighting factor, giving less weight to older data and more weight to recent data, thus promoting the estimation of slowly varying parameters [19]. Analogously to the OLS approach, the RLS algorithm can be implemented with the following set of recursive equations:

$$B_k = B_{k-1} + L_{k-1}(\mathbf{Y}_k - \Phi_k^T B_{k-1}) \tag{8}$$

where

$$L_k = P_{k-1}\Phi_k\left(\lambda + \Phi_k^T P_{k-1}\Phi_k\right)^T \tag{9}$$

and

$$P_k = \left(I - L_k\Phi_k^T\right)P_{k-1}\frac{1}{\lambda}. \tag{10}$$

Equations 8-10 have a structure similar to most recursive estimation schemes, as for example the Kalman Filter, with Equation 8 updating the parameters estimates at each step based on the error between the actual output and the modelled one. P_k is the usual covariance matrix, while L_k is the gain matrix. Finally, each element of μ_D can be obtained by replacing B_k into to (7). Additionally, the diagonal elements of Σ_D stem directly from matrix P_k of the corresponding regression.

4.2 | Gaussian processes

Gaussian Processes (GPs) are a non-parametric method that can be thought of a generalization of the Gaussian probability distribution to infinitely many variables. GPs are fully specified by a mean function $m(x)$ and covariance function $k(x, x')$:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{11}$$

In most applications there is no prior knowledge about the mean function, $m(x)$. Because GPs are, by definition, a linear combination of random variables with normal distribution, this is commonly assumed to be zero. The covariance function, $k(x, x')$, can be in general any function that takes any two arguments, such that $k(x, x')$ generates a non-negative definitive covariance matrix K. The covariance function implicitly specifies certain aspects of the process being modelled, such as smoothness and periodicity, among others. One of the frequently covariance function is the squared exponential, defined as:

$$k(x, x') = \sigma_f^2 exp\left(-\frac{1}{2l^2}|x - x'|^2\right). \tag{12}$$

It can be shown that using the squared exponential as a covariance function is equivalent to regression using infinitely many Gaussian shaped basis functions placed everywhere, and not just the training points [20].

In a learning problem GPs are then used to predict the output y_* given the test inputs x_*. Recalling that a Gaussian Process is a set of random variables which have a consistent Gaussian distribution with mean zero, we can represent such problem as:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right), \tag{13}$$

where σ_n is the variance of independent and identically distributed noise affecting the observations, and the different K matrix are built using any function $k(x, x')$ able to perform as a covariance function. Therefore, and considering the use of w training points, $K(X, X)$ will be an $w \times w$ matrix, $K(X, X_*)$ being an $1 \times w$ matrix and, naturally, $K(X_*, X)$ is $w \times 1$ matrix. As the number of training point w increases, computation of the different covariance matrix can become challenging.

Remembering that y and y_* are jointly Gaussian random vectors, then

$$y_*|y \sim \mathcal{N}(\bar{f}_*, cov(f_*)) \tag{14}$$

where

$$\bar{f}_* = k_*^T C_N^- 1 y \tag{15}$$

and

$$cov(f_*) = k_{**} - k_*^T C_N^{-1} k_* \tag{16}$$

The mean value of the prediction, $\bar{f}_*$ in Equation 15, gives the best estimate for y_*. Similarly, the variance $cov(f_*)$ is an indication of the prediction's uncertainty. In the Equations above $k_* = K(x, x_*)$, $C_N = K(X, X) + \sigma_n^2 I$ and $k_{**} = K(X_*, X_*)$, with σ_n being the variance of the noisy inputs.

The choice of the hyperparameters θ can play a relevant role in the prediction process. In the case of a covariance function given by (12), $\theta = \{l, \sigma_f, \sigma_n\}$. While the values for each element of θ could be made empirically, given some knowledge of the system, a more dynamic strategy could be adopted by maximizing the log likelihood of the training outputs given the inputs. The interested reader should refer to [20]. For more details on this.

Analogously to what was presented in the previous subsection, at each iteration of the DD-PF-GP independent GPs will be used for prediction. These individual GPs will be trained with historic data from the previous w time steps and yield the elements of μ_D, by using (15). Accordingly Σ_D will be a diagonal matrix, with the elements in the diagonal given by (16). Similarly to the DD-PF-LS, at time k the different matrix K in (13) are generated using for that the outputs of the filter from time instant $k - w$ to $k - 1$. The predicted mean and covariance matrix of the proposal distribution being estimated with then be the outputs of the GP regression, calculated using (14-16).

5 | NUMERICAL RESULTS

In this section the feasibility of the proposed DD-PF is analysed. The case under analysis considers a sensor-limited AUV under the influence of unmodelled disturbances. Such disturbances can be caused by lack of proper knowledge of the system, for example using sensors with unmodelled error sources, or by external sources, for example when the vehicle is subject to currents that cannot be measured. In what follows, only the horizontal position of the vehicle will be considered.

The simulations compare results obtained between the two proposed approaches, namely the DD-PF-LS and the DD-PF-GP, with two state-of-the-art filters being used for

underwater TBN, namely the SIR-PF and the PPF. The SIR-PF is not only the most widely used implementation of the PF framework, but also it has often been mentioned in the literature concerning TBN for sensor-limited AUVs. At the same time, the PPF has recently emerged as a robust solution for TBN problems.

The different set of simulations that will be presented are based on AUV trajectory with a constant surge velocity. For the simulated linear trajectories the heading of the vehicle was then considered to be always constant, while for the circular trajectories it was considered the heading to be varying at a constant rate. Such trajectories will then be affected by un-modelled disturbances. Simulated sonar measurements have been generated, at a rate of one per second, and consisting on a four beam sonar sensor, in a Janus configuration, similar to a DVL. The ranges returned from each of the beam were them corrupted with Gaussian noise. For that purpose, a synthetically generated bathymetric map was used. The different parameters being used in the simulations are shown in Table 1. For reference, the numerical results that are going to be presented are based on simulations performed on Matlab environment, and running on a laptop equipped with an Intel Core i3-2350M processor with 2 cores, and 4 giga byte of RAM memory. Additionally, and for the sake of simplicity the GPML Toolbox for Matlab [21] was used for implementing the GPs of the DD-PF-GP filter

In order to assess the convergence of the different filters to the true solution the ensemble Root Mean Square error (RMSE) of independent runs of the filters will be evaluated. The time-indexed RMSE metric for a two-dimension position, defined as follows, will be used:

$$RMSE_k = \sqrt{\frac{1}{M}\sum_{r=1}^{M}(\hat{x}_{k,r} - x_k)^2 + (\hat{y}_{k,r} - y_k)^2} \qquad (17)$$

where M is the number of independent MC runs of the filter. In what follows, M was set to one hundred. The RMSE was chosen for the sake of simplicity, but other metrics to compare performances of different PFs could have been used, like for example the Kullback-Leibler divergence [22].

Another feature of interest is the computational complexity of the filter. This is particularly relevant for sensor-limited systems, due to the limited computational power available on-board the vehicles. In order to compare the complexity of both DD-PF and SIR-PF two metrics can be used, namely the number of resampling steps performed, and the elapsed time for each run of the filter. These two metrics should provide an indication on the relative efficiency between each of the filters.

The number of required resampling steps is an interesting metric to consider, as it is known that the resampling stage represents an important share of the total computational complexity of PFs. At the same time it is the only step of the PF that cannot be run in parallel. However, the number of times a resampling algorithm is run does not provide any information on the increase in the global computation time of the filter. Despite such values being implementation dependent, they can be compared among each other, to assess the relative efficiency.

TABLE 1 Filter parameters used in the simulations

Filter Settings	
Number of Particles (N)	100
Process Noise (σ_w)	$\sqrt{5}m$
Measurement Noise (σ_v)	$1m$
Resampling Threshold (N_{thr})	$0.6N$
Initial Position (σ_{po})	$\sqrt{5}m$
Sensor Settings	
Number of Beams	4
Sensor Noise	$0.2m$
Learning Settings	
Forgetting Factor (λ)	0.9
Learning window (w)	10

5.1 | Linear trajectories

Initial simulations were performed to evaluate the performance of the DD-PF for a vehicle moving on a linear trajectory, but subject to external disturbances. Considering the vehicle surge velocity to be of 1 m/s^{-1}, the disturbances were set up to be of 0.3 m/s^{-1} in both X and Y directions. The trajectory performed by the vehicle can be seen in Figure 1. Additionally, these trajectories are overlaid on the contour levels of the topography of the bottom. It can be

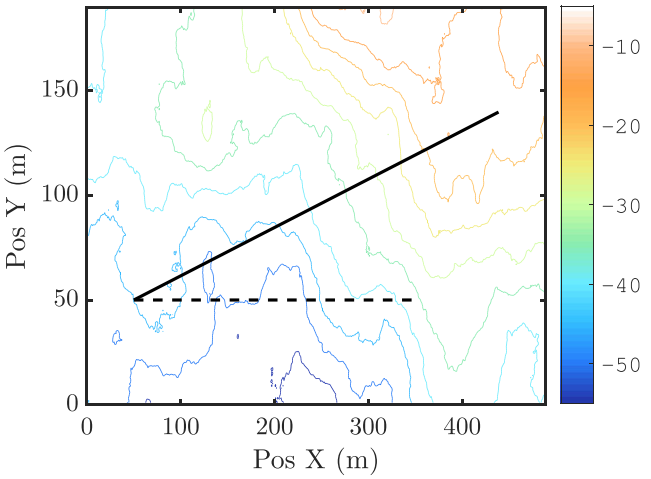

FIGURE 1 Simulated Linear Trajectories with the real trajectory (solid line), and the INS only trajectory (dashed) [Color figure can be viewed at wileyonlinelibrary.com]

seen that while the INS derived trajectory is only of 250 meters, the absolute error in position of the vehicle when subject to disturbances is of roughly 100 meters. This is equivalent to a disturbance affecting the vehicle of around 40% the distance travelled (DT), an already quite significant disturbance, and in line with some sensor-limited AUVs available.

Figure 2 illustrates the performance of each of the filter under analysis. It is noticeable the similar shape between the output of all the filters, highlighting the different levels of terrain information throughout the path followed by the vehicle. The SIR-PF, the PPF and the DD-PF-LS have similar performance, with their RMSE ranging below 3 meters. It should be noted that the DD-PF-LS achieved the lowest RMSE, of only 2.2 meters. At the same time, from these plots it is also possible to infer that the position outputs of the SIR-PF are particularly more noisy and less smooth than all the others. This can be relevant if, for example, the output of the TBN filter is going to be used as input to a main navigation filter, as suggested by some authors [23].

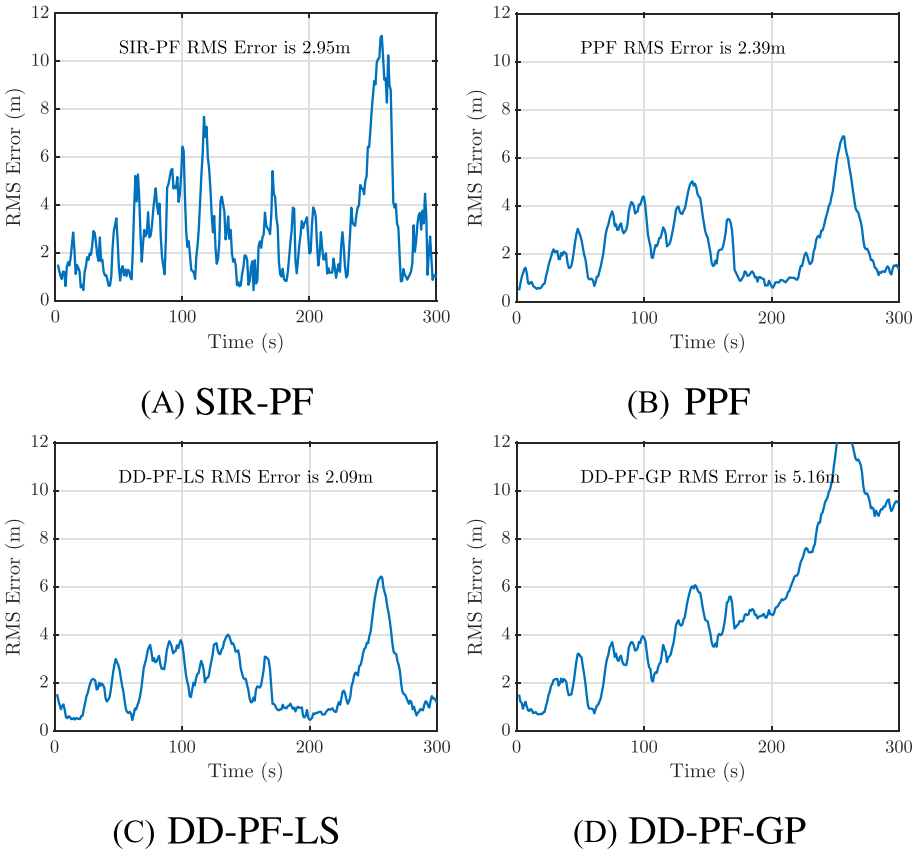

FIGURE 2 Comparison of the ensemble RMSE achieved by the filters for linear trajectories [Color figure can be viewed at wileyonlinelibrary.com]

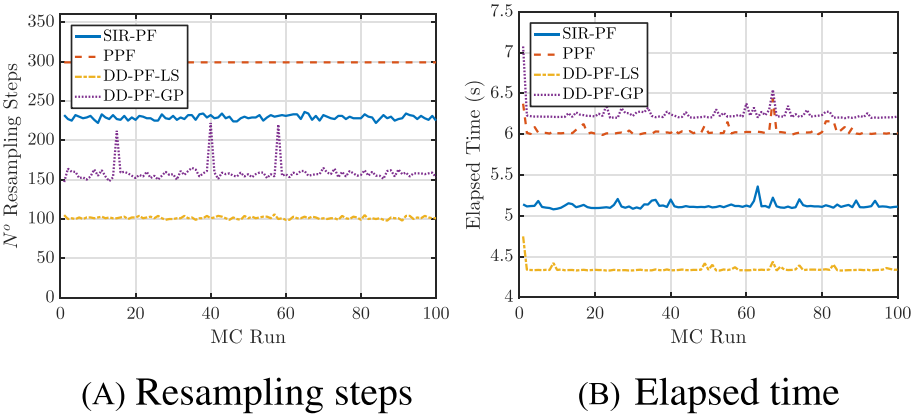

FIGURE 3 Comparison the complexity of the filters for linear trajectories [Color figure can be viewed at wileyonlinelibrary.com]

Figure 3 illustrates the complexity of the filters, by showing both the the number of resampling steps for each MC run, but also their elapsed time. From Figure 3a it is possible to conclude that both the DD-PF-LS and the DD-PF-GP require less resampling steps than the remaining. In fact, the DD-PF-LS is the most efficient one, performing resampling of its particles 40% less often than the standard SIR-PF, and 67% less often than the PPF. As for the DD-PF-GP, it also performs better when compared to the SIR-PF and the PPF. These are good indicators of the validity of the proposed approach. In line with these results, Figure 3b shows the time of batch processing each of the MC runs, for each of the filters. From there the main conclusion is that the DD-PF-LS is the filter running faster, requiring on average 17% less processing time than the SIR-PF. As expected, the PPF performs slightly slower than the SIR-PF, mostly due the additional number of resampling steps. However, perhaps not so surprisingly, the DD-PF-GP is the filter that is slower. This is justified by the GP learning framework, which requires additional computations.

5.2 | Circular trajectories

A second batch of simulations was performed, similar to the previous one but this time considering circular trajectories. Besides the change of the heading of the vehicle at a constant rate, all other conditions, including bottom topography and level of disturbances present, remained the same. The trajectory performed by the vehicle is illustrated in Figure 4.

Similarly to before, Figure 5 presents the ensemble RMSE of all the filters being considered. In these simulations the SIR-PF, with an ensemble RMSE of 2.7 meters, outperformed its competitor. Once again, the DD-PF-LS was more accurate than the DD-PF-GP, in this case by a small margin, while the PPF obtained the worst results, with an ensemble RMSE of 3.9 meters. While the SIR-PF performed better, as it obtained a smaller RMSE, it should be noted that the trajectories obtained by the DD-PF-LS are significantly smoother then all the others, which is a highly desirable characteristic as it suggests a steadier trajectory of the vehicle.

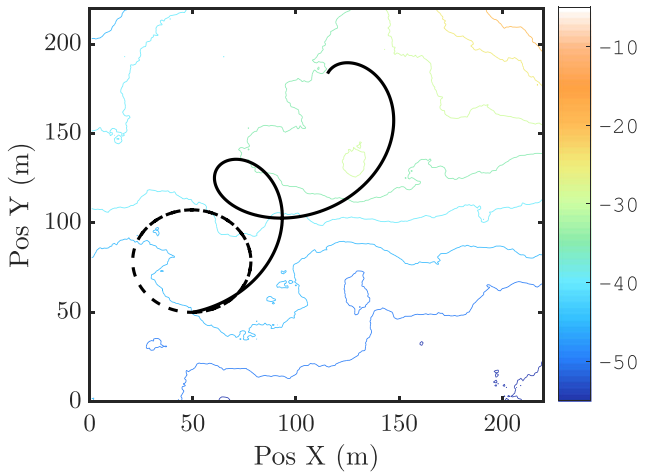

FIGURE 4 Simulated Circular Trajectories, with the real trajectory (solid line), and the INS only trajectory (dashed line) [Color figure can be viewed at wileyonlinelibrary.com]

As before, the complexity of the filters was also evaluated, with Figure 6a comparing the number of resampling steps per run for this batch of simulations. Similarly to before, the number of resampling steps required by both the DD-PF-LS and the DD-PF-GP is significantly less than for the SIR-PF. By comparing these numbers here obtained, with the ones obtained for linear trajectories, it can be noticed an increase in the number of resampling step for both the SIR-PF and the DD-PF-LS, with the DD-PF-GP performing resampling of its particle on approximately the same number of times as in the linear trajectories. Figure 6b details the elapsed time for each of the filters, and it presents some similarities with Figure 3b. Here again, the DD-PF-LS is the filter achieving better results, with lower processing time, demonstrating that increase of complexity related to the data-driven mechanism was more than compensated by requiring a smaller number of resampling steps.

5.3 | Low level of disturbances

The simulations presented so far address situations on which an AUV is following linear or circular trajectories, and subject to the effect of unmodelled disturbances at relatively high levels, that amount to approximately 40% of the distance travelled. This was in fact the main scenario behind the derivation of the two presented DD-PF algorithms. In what follows, we will study the behaviour of the aforementioned filters when the non-modelled disturbances are relatively smaller or even non-existent. For such analysis the simulations for linear trajectories were repeated, but now with smaller levels of disturbances. The purpose of such simulations is to study if the DD-PF also performs well when the process model is a more appropriate approximation of the true proposal density. Two different situations were assessed, one with disturbances amounting to 15% of the distance travelled, and another one without any disturbances. It should be noted that the process noise σ_w was maintained in the same levels as before. A summary of the obtained results can be found on Table 2, which shows the ensemble RMS and the average resampling steps of each filter, and for each simulated scenario. Additionally, and for an increased statistical significance, the standard deviation of these values is also provided.

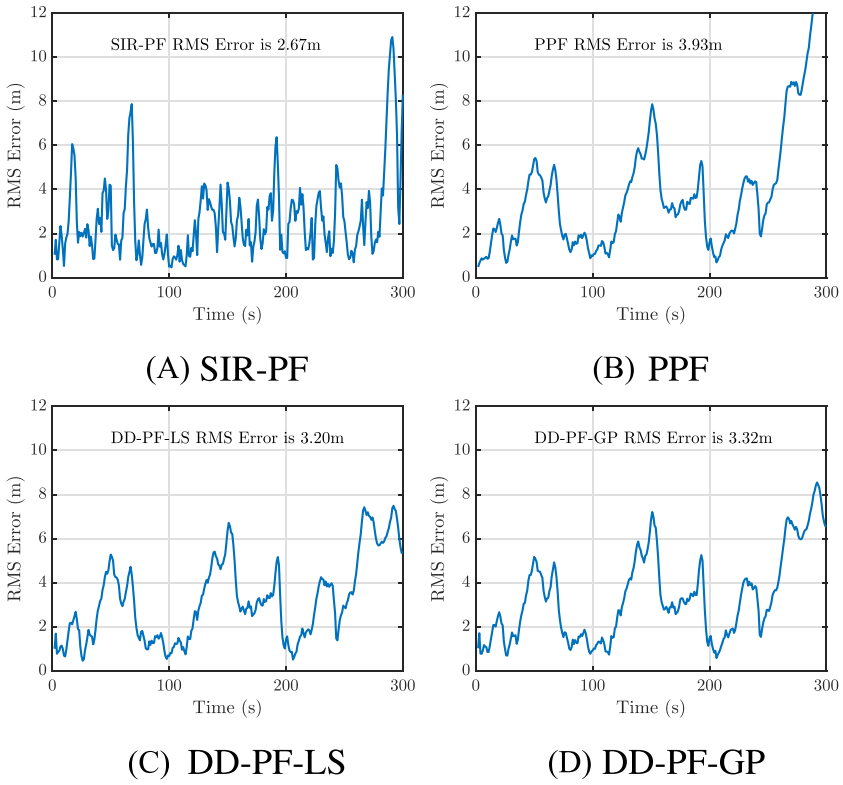

FIGURE 5 Comparison of the ensemble RMSE obtained by all the filters for circular trajectories [Color figure can be viewed at wileyonlinelibrary.com]

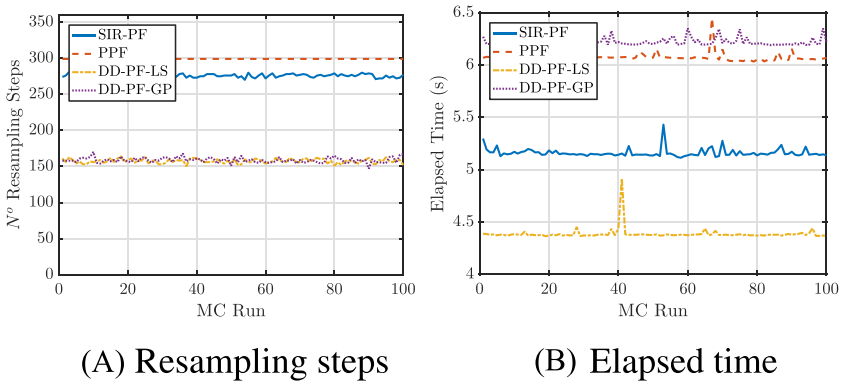

FIGURE 6 Comparison the complexity of the filters for circular trajectories [Color figure can be viewed at wileyonlinelibrary.com]

The obtained results are highlighted in Table 2. From there, it can be concluded that the DD-PF-LS performs relatively better than the SIR-PF an the PPF for all the foreseen scenarios, closely followed by the DD-PF-GP. At the same time, it is also more efficient, as the DD-PF-LS requires less resampling steps, and is processed in significantly less time. Interestingly, the SIR-PF also presents a higher standard deviation of its RMSE value, at least twice as much as the observed for the DD-PF-LS, which is also an indication of it's wobbly trajectory. Recalling from Figure 2, the RMSE plots for the PPF and the DD-PF-LS were much smoother than the ones for the SIR-PF, and a similar result is expected here. A close look to Table 2 reveals that the RMSE of the SIR-PF remains fairly constant when the disturbances are lower or even non-existent, which is counter-intuitive. However, this reflects the fact that the process noise was kept in the same levels for all the scenarios, which can be unadjusted and excessive when the disturbances are not so high.

In order to confirm that, the simulations were repeated, but this time adjusting the process noise to values compatible with the level of disturbances. A summary of the obtained results can be found on Table 3. As before, the tuning process for both the SIR-PF and the PPF consisted on adjusting the process noise to the lowest possible value, but while still being able to converge to the true trajec-

TABLE 2 Summary of the simulations for linear trajectories and different level of disturbances

Dist. (%DT)	SIR-PF		PPF		DD-PF-LS		DD-PF-GP	
	RMSE (std) (m)	Res. steps (std)	RMSE (std) (m)	Res. steps (std)	RMSE (std) (m)	Res. steps (std)	RMSE (std) (m)	Res. steps (std)
40%	2.95 (2.16)	229 (2.9)	2.39 (1.43)	299 (0.0)	**2.09** (1.31)	101 (1.1)	5.16 (3.16)	159 (11.1)
15%	2.62 (1.51)	233 (2.9)	1.15 (0.48)	299 (0.0)	**1.05** (0.47)	99 (1.1)	1.28 (0.48)	150 (3.1)
0%	2.95 (1.92)	247 (3.6)	1.03 (0.31)	299 (0.0)	**1.00** (0.35)	102 (1.2)	1.23 (0.34)	147 (3.4)

TABLE 3 Summary of the simulations for linear trajectories and different levels of disturbances with adjusted process noise

Dist. (%DT)	SIR-PF		PPF		DD-PF-LS		DD-PF-GP	
	RMSE (std) (m)	Res. steps (std)	RMSE (std) (m)	Res. steps (std)	RMSE (std) (m)	Res. steps (std)	RMSE (std) (m)	Res. steps (std)
40% ($\sigma_w = \sqrt{5}$)	2.95 (2.16)	229 (2.9)	2.39 (1.43)	299 (0.0)	**2.09** (1.31)	101 (1.1)	5.16 (3.16)	159 (11.1)
15% ($\sigma_w = \sqrt{1}$)	0.57 (0.31)	25 (0.8)	**0.38** (0.10)	150 (0.4)	0.78 (0.18)	296 (0.0)	0.69 (0.22)	296 (0.0)
0% ($\sigma_w = \sqrt{0.3}$)	**0.90** (0.45)	65 (0.9)	4.25 (2.38)	298 (0.5)	2.71 (1.5)	296 (0.0)	2.69 (1.5)	293 (1.6)

TABLE 4 Summary of the simulations for different number of particles

Number Particles	SIR-PF		PPF		DD-PF-LS		DD-PF-GP	
	RMSE (std) (m)	Res. steps (std)	RMSE (std) (m)	Res. steps (std)	RMSE (std) (m)	Res. steps (std)	RMSE (std) (m)	Res. steps (std)
100	2.95 (2.16)	229 (2.9)	2.39 (1.43)	299 (0.0)	**2.09** (1.31)	101 (1.1)	5.16 (3.16)	159 (11.1)
200	2.81 (2.19)	233 (2.4)	2.26 (1.42)	299 (0.0)	**2.00** (1.28)	104 (1.2)	2.10 (1.21)	176 (5.4)
500	2.73 (2.08)	235 (2.0)	2.19 (1.34)	299 (0.0)	**1.99** (1.21)	104 (1.0)	2.05 (1.49)	180 (5.5)

tory, for each particular setting of disturbance levels. The used values are also indicated on Table 3. As for both the DD-PF-LS ans the DD-PF-GP, the value for the variance of the process noise remained the same as of the simulations above.

As before, Table 3 shows the ensemble RMS and the average resampling steps of each filter, and for each simulated scenario, as well as the standard deviations of such values. From there the main conclusions are that when the disturbances are relatively mild or non-existent, the benefits of the proposed approach are less evident. It can be seen on Table 3 that with disturbances of around 15% of the distance traveled, and with adjusted process noise, the SIR-PF outperforms by a small margin the DD-PF-LS and the DD-PF-GP, but is the PPF that achieves a lower RMSE. However, the SIR-PF performs resampling on significantly less iterations than all the others, which is a good indicator of the lower complexity of the filters, as demonstrated in the previous subsection. As for the case when there are no disturbances present the SIR-PF performs significantly better than all the others, achieving a lower RMSE while at the same time being significantly more efficient. Furthermore, comparing the RMSE of the DD-PF in Tables 2 and 3 seems to indicate that the DD-PF is not as robust to large external disturbances as its counterparts.

5.4 | Number of particles

Following the simulations presented on the previous subsections, a final batch of simulations was performed to study the influence on the number for particles on the previous the presented results. To do so, the simulations of Section 5.1 were re-done, but now using a different number of particles. All the remaining parameters were not changed. The obtained results are summarized on Table 4, which compares the results when using 100 particles, presented previously, with the new results using 200 and 500 particles. As before, Table 4 presents both the ensemble RMS and the average number of resampling steps for each of the filters, but also the standard deviation of those values.

An increase on the number of particles seems to cause a small decrease on the observed RMSE for both the SIR-PF and the PPF. It should be noted however that by increasing the number of particles by a factor of five, only improved the RMSE by a meager 0.3 meters for the case of the SIR-PF, and 0.2 meters for the PPF. The increase on the number of particles seems to have a negligible effect on the RMSE of the DD-PF-LS. Thus, the overall tendency is that an increase of the number particles seems to cause a slight increase in this number, but not very significant.

For the case of the DD-PF-GP, increasing the number of particles from 100 to 200, did decreased the RMSE quite significantly, to less than half. However, further increasing the number of particles to 500 does not seem to yield

significantly better results. It is interesting to note that with an higher number of particles the DD-PF presents an RMSE which is lower than the SIR-PF, which seems to suggest that the DD-PF requires more particles, and thus more computational load, in order to perform better than the standard PF.

6 | CONCLUSIONS

In this article a new data-driven approach for the problem of underwater TBN was proposed. Differently from the standard SIR-PF, the DD-PF here presented tries to estimate the proposal distribution by learning from the data. This can be particularly advantageous when some of the disturbances present can't be modelled accurately. Two alternative learning methods have been proposed, the DD-PF-LS and the DD-PF-GP. Using a series of numerical examples the performance of the proposed filters was assessed, and compared to both the SIR-PF and the PPF. It has been demonstrated that under the effect of unmodelled strong disturbances the DD-PF-LS can outperform the usual SIR-PF and the PPF, achieving lower RMSE. On top of that, the DD-PF-LS has also been demonstrated to be significantly more efficient than both those filters, requiring less resampling steps and being processed faster. Additionally, the DD-PF-LS was also demonstrated to be more flexible, adapting to difference levels of disturbances with only small variatons on its performance. On the other hand, the simulations presented show that the DD-PF-GP always takes requires more time, mostly due to the use of a computationally expensive GP framework. Moreover, the DD-PF-GP was unable to cope with high levels of disturbance, even though it performed remarkably well for the other analysed scenarios.

While the simulations for the DD-PF focused only on a two-dimensional position estimation problem, the DD-PF is still applicable for situations requiring a complete navigation solution, including estimation of the full pose of the vehicle. In such situations, the DD-PF could be used, for example, to generate position updates to a main navigation filter, as for example suggested in 23. Future work would include a further study on the use of GPs as a learning methods, perhaps investigating alternative mean and covariance functions. At the same time, it would be interesting to assess if the use of such nonparametric algorithm could be more appropriate for a wider range of application scenarios.

ACKNOWLEDGEMENTS

This work is financed by the ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme, and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project ≪ POCI-01-0145-FEDER-006961 ≫.

ORCID

José Melo https://orcid.org/0000-0001-5190-9730
Aníbal Matos https://orcid.org/0000-0002-9771-002X

REFERENCES

1. U. Jorgensen and R. Skjetne, *Online reconstruction of drifting underwater ice topography: The 2d case*, Asian J. Control **17** (2015), no. 5, 1509–1521.
2. C. Morice, S. Veres, and S. McPhail, *Terrain referencing for autonomous navigation of underwater vehicles*, Proc. of of the MTS/IEEE Oceans'09 Conf., IEEE, Bremen, Germany, 2009, pp. 1–7.
3. G. T. Donovan, *Position error correction for an autonomous underwater vehicle inertial navigation system (INS) using a particle filter*, IEEE J. Ocean. Eng. **37** (2012), no. 3, 431–445.
4. D. K. Meduna, S. M. Rock, and R. S. McEwen, *Low-cost terrain relative navigation for long-range AUVs*, Proc. of of MTS/IEEE Oceans'08 Conf., IEEE, Quebec City, Canada, 2008, pp. 1–7.
5. J. Melo and A. Matos, *Survey on advances on terrain based navigation for autonomous underwater vehicles*, Ocean Eng. **139** (2017), 250–264.
6. K. B. Anonsen and O. K. Hagen, *Recent developments in the HUGIN AUV terrain navigation system*, Proc. of of the MTS/IEEE Oceans'11 Conf., IEEE, Waikoloa, HI, USA, 2011, pp. 1–7.
7. F. C. Teixeira, A. Pascoal, and P. Maurya, *A novel particle filter formulation with application to terrain-aided navigation*, 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles, Vol. **45**, Porto, Portugal, 2012, pp. 132–139.
8. T. Zhou et al., *Adaptive particle filter based on kullback-leibler distance for underwater terrain aided navigation with multi-beam sonar*, IET Radar Sonar Navig. **12** (2018), 433–441(8).
9. J. F. G. de Freitas et al., *Sequential Monte Carlo methods to train neural network models*, Neural Comput. **12** (2000), no. 4, 955–993.
10. R. van der Merwe et al.: *The unscented particle filter*. CUED/F-INFENG/TR 380. Cambridge University Engineering Department, 2000.
11. B Ju, Z Zhang, and J Zhu, *A novel proposal distribution for particle filter*, 3rd Int. Congress on Image and Signal Processing, Vol. **7**, 2010, pp. 3120–3124.
12. F. C. Teixeira et al., *Robust particle filter formulations with application to terrain-aided navigation*, Int. J. Adapt Control **31** (2017), no. 4, 608–651.
13. M. Lager, E. A. Topp, and J. Malec, Underwater terrain navigation during realistic scenarios, *Multisensor Fusion and Integration in the Wake of Big Data, Deep Learning and Cyber Physical System*, Springer International Publishing, Cham, 2018, pp. 186–209.
14. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, Cambridge, MA, USA, 2005.
15. M. S. Arulampalam et al., *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*, IEEE Trans. Signal Process. **50** (2002), no. 2, 174–188.

16. A. Murangira et al., *Robust regularized particle filter for terrain navigation*, 14th International Conf. on Information Fusion, Chicago, IL, USA, 2011, pp. 1–8.
17. D. Crisan and A. Doucet, *A survey of convergence results on particle filtering methods for practitioners*, IEEE Trans. Signal Process. **50** (2002), no. 3, 736–746.
18. X. Hu, T. B. Schon, and L. Ljung, *A general convergence result for particle filtering*, IEEE Trans. Signal Process. **59** (2011), no. 7, 3424–3429.
19. A. Vahidi, A. Stefanopoulou, and H. Peng, *Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments*, Veh. Syst. Dyn. **43** (2005), no. 1, 31–55.
20. C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, The MIT Press, Cambridge, 2006.
21. C. E. Rasmussen and H. Nickisch, *Gaussian processes for machine learning (gpml) toolbox*, J. Mach. Learn. Res. **11** (2010), 3011–3015.
22. R. Chou et al., *Performance evaluation for particle filters*, 14th International Conf. on Information Fusion, Chicago, IL, USA, 2011, pp. 1–7.
23. D. K. Meduna, S. M. Rock, and R. S. McEwen, *Closed-loop terrain relative navigation for AUVs with non-inertial grade navigation sensors*, IEEE/OES Autonomous Underwater Vehicles, IEEE, Monterey, CA, USA, 2010, pp. 1–8.

AUTHOR BIOGRAPHIES

José Melo is now a MCM scientist at the NATO Centre for Maritime Research and Experimentation (CMRE). Before, he was a researcher at INESC TEC, in Porto, Portugal. In 2016 he received his PhD in Electrical and Computer Engineering from the Faculty of Engineering, University of Porto. In 2008 he received his M.Sc. degree in Electrical and Computer Engineering, with a major in Automation, from the same University. In between degrees, and until 2010, José has worked as a Software Engineer, at CERN. His current research interests are in the area of Navigation and Sensor Fusion for Autonomous Vehicles.

Aníbal Matos received a PhD in Electrical and Computer Engineering form Porto University in 2001. He is currently coordinator of the Centre for Robotics and Autonomous Systems at INESC TEC and also an assistant professor at the Faculty of Engineering of Porto University. His main research interests are related to perception, sensing, navigation, and control of autonomous marine robots, being the author or co-author of more than 100 publications in international journals and conferences. He has participated and lead several research projects on marine robotics and its application to monitoring, inspection, search and rescue, and defense.

How to cite this article: Melo J, Matos A. A data-driven particle filter for terrain based navigation of sensor-limited autonomous underwater vehicles. *Asian J Control*. 2019;1–12. https://doi.org/10.1002/asjc.2107