Adaptive Approximate Bayesian Computational Particle Filters for Underwater Terrain Aided Navigation

Camille Palmier, Karim Dahia, Nicolas Merlinge, Pierre del Moral, Dann Laneuville, Christian Musso

Adaptive Approximate Bayesian Computational Particle Filters for Underwater Terrain Aided Navigation

Camille Palmier*, Karim Dahia*, Nicolas Merlinge*,
Pierre Del Moral**, Dann Laneuville*** and Christian Musso*

* ONERA - The French Aerospace Lab
** Inria
*** Naval Group

***Abstract*—To perform long-term and long-range missions, underwater vehicles need reliable navigation algorithms. This paper considers multi-beam Terrain Aided Navigation which can provide a drift-free navigation tool. This leads to an estimation problem with implicit observation equation and unknown likelihood. Indeed, the measurement sensor is considered to be a numerical black box model that introduces some unknown stochastic noise. We introduce a measurement updating procedure based on an adaptive kernel derived from Approximate Bayesian Computational filters. The proposed method is based on two well-known particle filters: Regularized Particle Filter and Rao-Blackwellized Particle Filter. Numerical results are presented and the robustness is demonstrated with respect to the original filters, yielding to twice as less non-convergence cases. The proposed method increases the robustness of particle-like filters while remaining computationally efficient.**

I. INTRODUCTION

The ability of an underwater vehicle (UV) to accomplish a mission depends on the performance of on-board navigation algorithms. This paper considers Terrain Aided Navigation (TAN) which can provide a drift-free navigation tool for UV, yielding a powerful alternative to current navigation methods which include resurfacing for GPS [8]. TAN generates vehicle position estimates by correlating terrain measurements obtained by a multi-beam sensor together with stored terrain maps, which is related to a filtering problem.

The aim of filtering is to estimate the state of an evolving system, customarily modeled by a stochastic process and called the state process. The state process cannot be measured directly but only via a related process named the observation process. The filtering problem consists in computing the posterior density of the state at the current time given the observation data accumulated up to that time. This state-space problem can be solved analytically when the propagation of the state process and the observation process are both linear and when the noises are independent white Gaussian. In this case, the Kalman Filter (KF) is known to be optimal. The KF was extended to non-linear models, but may become highly unstable in case of non-Gaussian processes or severe non-linearities such as multimodalities, i.e. multiple maxima, also called modes, in the posterior density.

This paper considers higly non-linear and non-Gaussian observation processes. Indeed, the terrain profile generates severely non-linear and ambiguous measurements, which yields multimodalities. The Kalman-like filters cannot be used for this class of problems. To tackle non-linearities such as multimodalities, particle filtering methods were introduced [3]. Particle filters approximate the posterior density by a mixture of weighted Dirac functions called particles.

The present work addresses the case where the noise of the observation process (also called measurement noise) is unknown. The observation model is the result of a numerical black box model, therefore the likelihood is unknown. Indeed, in the case of underwater TAN using a multi-beam bathymetric sensor, the observation equation is implicit and requires a numerical approximation in practice. The stochastic characteristics of the error between the actual measurements and the numerical approximation of the observation cannot be inferred.

To address cases where the measurement noise distribution is unknown (therefore the likelihood is unknown), Approximate Bayesian Computational (ABC) methods were introduced ([2], [12], [6]). ABC is a class of estimation methods used to overcome the lack of knowledge about the measurement distribution. This method bypasses the evaluation of the likelihood function. An acceptance/rejection test on a uniform law is applied to the samples state. When a candidate sample state is rejected, it is drawn again until it is accepted. This algorithm is computationally demanding since it leads to a rise of the number of samples. To reduce the computational load of ABC methods, a filtering approach was derived - called ABC filter. ABC filter replaces the measurement likelihood by an analytical density function - called kernel - whose bandwidth is adaptive [6]. This scale parameter aims to account for every state density's mode while the measurement ambiguity is not removed. However, it is difficult to tune, which may yield filter unstability.

We propose a new way of choosing this setting, based on the efficiency degeneracy criterion [4]. This criterion is commonly used in particle filters to trigger the resampling step. The proposed method is called Adaptive Approximate Bayesian Computation (A2BC) and can be integrated in any particle-like filter. Two filters have been selected in this work: Regularized Particle Filter (RPF [9]) and Rao-Blackwellized

Particle Filter (RBPF [13]).

The general formulation of the estimation problem, Particle Filter and ABC filter approaches are recalled in Section II. In Section III, we describe the principle of the A2BC method and its implementation within conventional particle filters. In Section IV, we present the problem of submarine TAN navigation. After describing the performance criteria, we numerically compare the performance of two A2BC filters with these of conventional particle filters. Finally, we will conclude in Section V.

II. Problem statement

A. Filtering problem and particle filter

Consider the following discrete-time state-space model with hidden states $\{X_k\}_{k\in\mathbb{N}}$ and observations $\{Y_k\}_{k\in\mathbb{N}}$ given by

$$X_k = f_k(X_{k-1}, \eta_k) \tag{1}$$
$$Y_k = h_k(X_k, Y_k, \nu_k) \tag{2}$$

where f_k and h_k are possibly non-linear functions and η_k and ν_k are independent white noises. The underwater application motivates the choice of an implicit observation model [8].

We shift from this implicit formulation (2) of the observation process to an explicit model through numerical approximations: the observation model is the result of a numerical black box code. The function h_k is injective: when X_k is fixed, only one Y_k satisfies equation (2). The implicit problem can therefore be solved by using the fixed point method or sample approximation methods. Thus, we change the observation model (2) by

$$Y_k = h'_k(X_k, \nu'_k) \tag{3}$$

where h'_k is still a possibly non-linear function and where ν'_k is a new measurement noise. Noise ν'_k is different from ν_k. Indeed, the stochastic characteristics of the error between the actual measurements (2) and the numerical approximation (3) of the observation cannot be inferred. ν'_k is divided into two parts: one part is derived from the numerical approximation and the other is due to an additional measurement noise. The measurement sensor is the output of a numerical code which is not available, thus the likelihood cannot be computed.

We want to estimate the posterior density

$$p_k(x) \triangleq p(X_k = x|Y_{1:k}) \tag{4}$$

where $Y_{1:k} = [Y_1, Y_2, ..., Y_k]$ is the vector of all the observation data accumulated up to the time k.

The state estimation consists of two steps: prediction and correction.

- The prediction step determines a predictive distribution $p(X_k|Y_{k-1})$ with respect to the dynamical model uncertainty $p(X_k|X_{k-1})$ and the previous posterior distribution $p(X_{k-1}|Y_{k-1})$ via the Chapman-Kolmogorov equation:

$$p(X_k|Y_{k-1}) = \int p(X_k|X_{k-1})p(X_{k-1}|Y_{k-1})dX_{k-1} \tag{5}$$

- The correction step determines the posterior distribution of the state with respect to the predictive distribution (5) and the likelihood $p(Y_k|X_k)$. From Bayes' law, one obtains:

$$p(X_k|Y_k) = \frac{p(X_k|Y_{k-1})p(Y_k|X_k)}{\int p(X_k|Y_{k-1})p(Y_k|X_k)dX_k} \tag{6}$$

The particle filter estimates the posterior density by means of a Dirac mixture of N weighted particles (X_k^i, w_k^i):

$$\widehat{p}_k(x) \triangleq \sum_{i=1}^{N} w_k^i \delta_{x=X_k^i} \tag{7}$$

There are a multitude particle-based algorithms in the literature. We focuse on two of them, which are known for their robustness to non-linearities: the Regularized Particle Filter (RPF) ([9], [10]) and the Rao-Blackwellized Particle Filter (RBPF) [13]. The particularity of RPF is the use of kernel density estimation sampling, in order to smooth the state density estimate. RPF is particularly useful when the process density is narrow. RBPF takes advantage of the structures of the state process by setting aside a linear part that will be solved by a Kalman filter (which is optimal if the processes are Gaussian). Using partially optimal filter allows the filter's variance to be smaller and reducing the particle state vector size limits the computational load. When the likelihood $p(Y_k|X_k)$ is unknown, the particle filter correction step cannot be explicitly achieved. To tackle this issue, ABC methods were introduced [2].

B. ABC methods

ABC filters are useful when the likelihood is unknown. These methods match observations with simulated pseudo-observations and therefore avoids to assess the likelihood ([2], [12], [6]). The objective is to evaluate the particle weights as a function of the distance between the actual observation and the simulated pseudo-observations.

The posterior density $p_k(x)$ is derived by marginalizing the joint density $p(X_{0:k}|Y_{1:k})$. We can get X_k^i samples obtained from (1). From these samples, we generate N pseudo-observations, $U_k^i = \mathcal{M}(X_k^i)$ where $\mathcal{M}$ is the numerical model. The model $\mathcal{M}$ is the h' function (3) without the additional measurement noise included in ν'_k. The closeness between the pseudo-observations U_k and the true observed Y_k is determined by a kernel probability density function $K_\epsilon(Y_k, U_k^i)$. The posterior density is estimated by:

$$p(X_{0:k}|Y_{1:k}) = p(X_0) \int p(X_{1:k}, U_{1:k}|Y_{1:k})dU_{1:k} \tag{8}$$

$$p(X_{0:k}|Y_{1:k}) \propto$$
$$p(X_0) \prod_{k\geq 1} \left[\int K_\epsilon(Y_k, U_k) p(U_k|Y_k, X_k) dU_k \right] p(X_k|X_{k-1}) \tag{9}$$

The ABC importance weights update is as follows:

$$w_k^i = w_{k-1}^i K_\epsilon(Y_k, U_k^i) \tag{10}$$

The choice of the kernel scale parameter ϵ has a strong impact on estimation convergence. It needs to be chosen in an adaptive manner, otherwise the filter may abruptly fail if the true observation at one time Y_k falls in distribution tails. In the following sub-section, we propose an approach to choose this parameter.

III. Adaptative Approximate Bayesian Computational Particle Filter (A2BC-PF)

A. A2BC

The scale parameter aims to account for every state density's modes while the measurement ambiguity is not removed. The loss of these modes leads to the degeneration of the filter weights. The parameter ϵ will be chosen adaptively in order to robustify ABC methods.

We introduce a method to determine ϵ based on the efficiency degeneracy criterion [4]:

$$N_{eff} = \frac{1}{\sum_{i=1}^N {w_k^i}^2} \tag{11}$$

where w_k^i are the particle weights and N the number of particles. This criterion is commonly used in particle filters to trigger the resampling step. Resampling is initiated whenever $N_{eff} < N_{th}$ where $N_{th} = \theta N$ and $0 < \theta \leq 1$ is a given threshold.

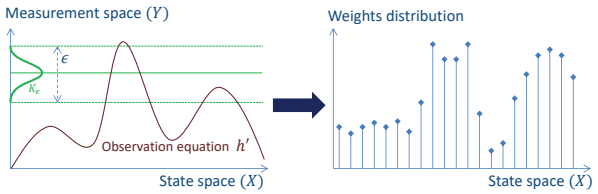
Since the likelihood is unknown, the ABC filter method is used to compute particle weights. From (10), the efficiency criterion can be written as:

$$N_{eff}(\epsilon_k) = \frac{1}{\sum_{i=1}^N (w_{k-1}^i K_{\epsilon_k}(Y_k, U_k^i))^2} \tag{12}$$

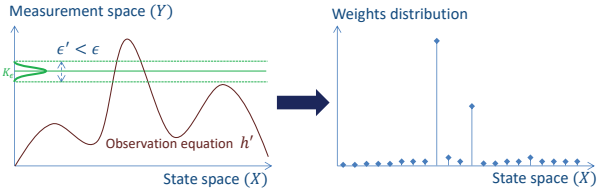
The idea is to choose $\epsilon \in I$ such that the efficiency criterion (12) is greater than the resampling threshold. At each time-step, an optimal $\hat{\epsilon}_k$ is determined by solving the following problem:

$$\hat{\epsilon}_k = \underset{\epsilon \in I}{\operatorname{argmin}} |N_{eff}(\epsilon) - N_{th}| \tag{13}$$

We want to avoid the case where the distribution of weights tends towards a Dirac distribution (i.e. when a single weight tends to unity and all the others tend to zero, see Figure 1). In order to keep the information, the case where epsilon is large is also avoided otherwise neither particle is favored. The scale parameter ϵ must be such that the efficiency criterion is equal or just above the resampling threshold. Since $\epsilon \in I$, the solution $\hat{\epsilon}$ to (13) is not guaranteed to yield N_{eff} greater than N_{th}. In case of $N_{eff} < N_{th}$, a conventional resample step is triggered to prevent from the filter degenerency. The kernel bandwidth has a strong impact on the degeneracy of the weights distribution, which may cause some estimation divergence. In practice, finding $\hat{\epsilon}$ (13) can be solved by making numerical approximations (e.g. gradient descent or grid search).



(a) Correction step using a given kernel bandwidth epsilon and the subsequent weight distribution.



(b) If the kernel bandwidth is too narrow, the degeneracy phenomenon arrises.

Fig. 1: Relation between the scale parameter of the kernel and the weights distribution.

B. A2BC-RPF and A2BC-RBPF

In this section, two improved Particle Filter are derived from the conventional RPF and RBPF, so that their correction steps satisfy (13).

► RPF ([10], [9]) is based on the kernel estimation theory [14]. After the resampling step is triggered, the posterior density defined by a mixture of Dirac functions (7) can be rewritten as a mixture of weighted kernels:

$$\hat{p}_k(x) = \sum_{i=1}^N w_k^i K_h(X_k - X_k^i) \tag{14}$$

When all the particles have the same weight, an optimal kernel K and an optimal bandwidth h_{opt} can be determined by minimizing the Mean Intergrater Square Error criterion [14], [9]. This additional step is called regularization, see Algorithm 1.

► RBPF is used when the state vector X_k can be decomposed into two sub-vectors (X_k^n, X_k^l) such that the model is linear with respect to X_k^l conditionally to X_k^n and $Y_{0:k}$:

$$X_{k+1}^n = f_k^n(X_k^n) + F_k^n(X_k^n)X_k^l + G_k^n(x_k^n)\eta_k^n \tag{15}$$
$$X_{k+1}^l = f_k^l(X_k^n) + F_k^l(X_k^n)X_k^l + G_k^l(x_k^n)\eta_k^l \tag{16}$$
$$Y_k = h_k(X_k^n) + H_k(X_k^n)X_k^l + \nu_k \tag{17}$$

where $\eta_k = [\eta_k^n, \eta_k^l]^T$ is a Gaussian noise with covariance matrix $Q_k = \begin{pmatrix} Q_k^n & Q_k^{ln} \\ (Q_k^{ln})^T & Q_k^l \end{pmatrix}$, v_k is Gaussian with covariance

Algorithm 1 A2BC-RPF Algorithm

Initialization: For $i = 1, ..., N$, initialize the particles $X_0^i \sim p(X_0)$ from a prior distribution and set $w_{-1}^i = 1/N$.

For $k \geq 0$ **do**

1) ABC correction:
   - For $i = 1, ..., N$, the pseudo-observations are $U_k^i = \mathcal{M}(X_k^i)$.
   - Determine $\hat{\epsilon}_k$ such that $\hat{\epsilon}_k = \text{argmin}_{\epsilon_k \in I} |N_{eff}(\epsilon_k) - N_{th}|$
   - Update the weights $\tilde{w}_k^i = w_{k-1}^i K_{\hat{\epsilon}_k}(Y_k, U_k^i)$
   - Normalization $w_k^i = \frac{\tilde{w}_k^i}{\sum_i \tilde{w}_k^i}$
2) Compute the estimate $\hat{X}_k = \sum_i w_k^i X_k^i$
3) Compute the empirical covariance matrix $\hat{P}_k = \sum_i w_k^i (X_k^i - \hat{X}_k)(X_k^i - \hat{X}_k)^T$
4) **If** $N_{eff} < N_{th}$ **do**
   - Apply one of the resampling procedure (see [5] for a review on resampling methods). Discard/multiply particles X_k^i according to high/low weights w_k^i and denote by X_k^i the selected states. Set $w_k^i = 1/N$.
   - Regularization step:
     - Compute D_k such that $D_k D_k^T = \hat{P}_k$
     - Draw ζ^i from a kernel
     - $X_k^i = X_k^i + h_{opt} D_k \zeta^i$

   end
5) Prediction step: For $i = 1, ..., N$, sample $X_{k+1}^i \sim p(X_{k+1}|X_k^i)$ see (1).

end

matrix R_k, the nonlinear part follows the known law $p(X_0^n)$ and the linear part has a Gaussian density $\mathcal{N}(X_0^l, P_0)$, see Algorithm 2. The posterior density can be written as follows:

$$p(X_k^l, X_{0:k}^n | Y_{0:k}) = p(X_k^l | X_{0:k}^n, Y_{0:k}) p(X_{0:k}^n | Y_{0:k}) \quad (18)$$

Under the hypothesis that $p(X_k^l | X_{0:k}^n, Y_{0:k})$ is Gaussian, this density is estimated by a Kalman filter. The non-linear and multimodes density $p(X_{0:k}^n | Y_{0:k})$ is estimated by a particle filter. The linear part follows a Gaussian law, so it is sufficient to calculate at each time step its average and its covariance by using the Kalman filter (see [13]). The particle prediction will be given by:

$$p(X_{0:k}^n | Y_{0:k-1}) = p(X_k^n | X_{0:k-1}^n, Y_{0:k-1}) p(X_{0:k-1}^n | Y_{0:k-1}) \quad (19)$$

This density is Gaussian with mean and covariance matrix given by equation (25) in [13]. The resulting approximation is given by:

$$\hat{p}_k(x) = \sum_{i=1}^{N} w_k^i \mathcal{N}(x^l; X_k^{l,i}, P_k^i) \delta_{x^n = X_k^{n,i}} \quad (20)$$

where P_k^i is the covariance matrix of the linear part $X_k^{l,i}$.

Algorithm 2 A2BC-RBPF Algorithm

Initialization: For $i = 1, ..., N$, sample $X_0^{n,i} \sim p(X_0^n)$ and set $\{X_{0|-1}^{l,i}, P_{0|-1}^i\}_{i=1:N} = \{X_0^l, P_0^l\}$ and $w_{-1}^i = 1/N$.

For $k \geq 0$ **do**

1) ABC correction:
   - For $i = 1, ..., N$, the pseudo-observations are $U_k^i = \mathcal{M}(X_k^{n,i}, X_{k|k-1}^{l,i})$
   - Determine $\hat{\epsilon}_k$ such that $\hat{\epsilon}_k = \text{argmin}_{\epsilon_k \in I} |N_{eff}(\epsilon_k) - N_{th}|$
   - Update the weights $\tilde{w}_k^i = w_{k-1}^i K_{\hat{\epsilon}_k}(Y_k, U_k^i)$
   - Normalization $w_k^i = \frac{\tilde{w}_k^i}{\sum_i \tilde{w}_k^i}$
2) Compute the estimate $\hat{X}_k^n = \sum_i w_k^i X_k^{n,i}$
3) Compute the empirical covariance matrix $\hat{P}_k^n = \sum_i w_k^i (X_k^{n,i} - \hat{X}_k^n)(X_k^{n,i} - \hat{X}_k^n)^T$
4) **If** $N_{eff}(\hat{\epsilon}_k) < N_{th}$ **do**
   - Discard/multiply particles $X_k^i = [X_k^{n,i}, X_{k|k-1}^{l,i}]$ according to high/low weights w_k^i and denote by X_k^i the selected states. Set $w_k^i = 1/N$.
   - Regularization step:
     - Compute D_k such that $D_k D_k^T = \hat{P}_k$ where $\hat{P}_k = diag([\hat{P}_k^n, P_{k|k-1}])$.
     - Draw ζ^i from a kernel
     - $X_k^i = X_k^i + h_{opt} D_k \zeta^i$

   end
5) Kalman correction: compute $X_k^{l,i}$ and P_k^i according to equation (22) in [13].
6) Particle prediction: sample $X_{k+1}^{n,i}$ using (19) (see equation (25) in [13]).
7) Kalman prediction: compute $X_{k+1|k}^{l,i}$ and $P_{k+1|k}^i$ according to equation (23) in [13].

end

A regularization step is added in order to increase the robustness of the A2BC-RBPF filter. For a same number of particles, RBPF is more computationally intensive since, it requires one Kalman filter per particle. However, it is possible to obtain a compromise between calculation cost and performance in favour of the RBPF for many applications. For example, in TAN, the Kalman filter covariance matrix is independent of the non-linear part, which reduces the complexity of the RBPF since there is only one covariance matrix to update.

IV. Simulation example

To illustrate the behavior of the resulting A2BC filters, an application to underwater Terrain Aided Navigation (TAN) is presented. The observation equation is described and numerical results are presented.

A. Underwater terrain aided navigation

TAN can provide a drift-free navigation tool for underwater vehicles (UV), yielding a powerful alternative to current navigation methods which include resurfacing for GPS. TAN

generates vehicle position estimates by correlating terrain measurements obtained by a multi-beam sensor (see Figure 2) together with stored terrain maps.

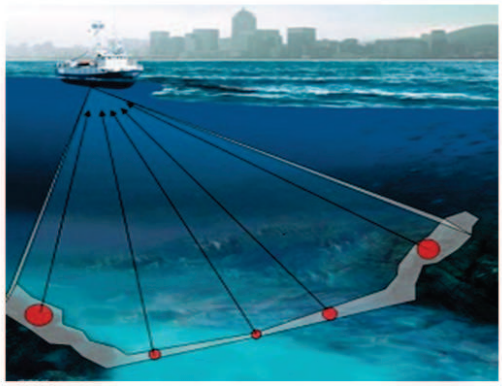


Fig. 2: Multi-beam sensor.

Define $X = [P, V]^T$ the state vector composed of $P = [p_x, p_y, p_z]^T$ the position vector expressed in meters and $V = [v_x, v_y, v_z]^T$ the vector of velocities expressed in meter per second. A simple case of state dynamics is chosen, known as NCV model [1]. The dynamical model is written in the following discrete way:

$$\begin{cases} P_k = P_{k-1} + \Delta t V_{k-1} + \eta_k^X \\ V_k = V_{k-1} + \eta_k^V \end{cases} \tag{21}$$

where Δt is the discretization time-step and $\eta_k = [\eta_k^X, \eta_k^V]^T$ is the state noise.

The measurement is made of m beams: $Y = [r_1, ..., r_m]^T$. Each measurement r_i returns the distance between the UV and the seabed (see Meduna [7] for details):

$$r_i = \sqrt{(p_x - p_x^i)^2 + (p_y - p_y^i)^2 + (p_z - h(p_x^i, p_y^i))^2} + \nu_i \tag{22}$$

where h is the terrain depth. The terrain function is the Digital Elevation Model (DEM). In our case, DEM is defined by a regular grid of elevation values, see Figure 3. The spatial resolution of the chosen map is about 100 meters. The intersection point of beam r_i direction vector with the terrain is denoted $(p_x^i, p_y^i, p_z^i = h(p_x^i, p_y^i))$. Since the coordinates of the intersection point depend on the beam vector r_i, the measurement model is implicit. The measurement equation (22) has to be rewritten as (3). There are several ways to get an explicit formula, e.g. ray tracing, grid search. In this paper, the explicit formula is the output of a numerical model named $\mathcal{M}$, that introduces some unknown sampling noise.

The law of this noise cannot be easily inferred which motivate the use of the method developed in Section III.

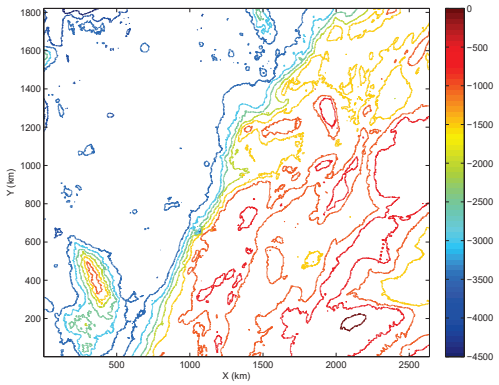


Fig. 3: Map of the contours lines of the California coast (31 ° 88' N, 121 ° 27' W). The colorbar represents the depth levels in meters.

B. Performance criteria

In order to compare the algorithms, we use the following criteria, evaluated for $N_{mc} \in \mathbb{N}$ Monte-Carlo simulations:

- The Root Mean Square Error (RMSE):

$$RMSE_k = \sqrt{\frac{\sum_{i=1}^{N_{mc}} (\widehat{X}_k^i - X_k)^T (\widehat{X}_k^i - X_k)}{N_{mc}}} \tag{23}$$

  where $\widehat{X}_k$ is the kth estimated of the state. We will compare these RMSEs with the Posterior Cramér Rao Bound (PCRB) which is calculated according to the Tichavský recursive formula [15]. The PCRB is approximated over 100 state samples at each time-step.

- The number of non-convergence:
  The filter is said to not converge if, at the end of the trajectory, during the last 5 consecutive measurement times, the state estimate $\widehat{X}_k$ leaves the confidence ellipsoid Γ_k given by the PCRB, such that

$$\Gamma_k = \{X_k | (X_k - \widehat{X}_k)^T PCRB_k^{-1} (X_k - \widehat{X}_k) \leq \alpha^2\} \tag{24}$$

  where the threshold α is equal to the probability $\mathbb{P}(\mathcal{X}^2(d) \leq \alpha^2) = 0.99$ with d the dimension of the state vector.

C. Simulation and results

The kernel in (10) is chosen as a Cauchy function whose scale parameter ϵ is fixed for conventional filters and determine adaptively with A2BC filters. The linear part of the state in RBPF is the velocity.

For 100 Monte Carlo trials, the number of non-convergence is shown in Table I. The initial number of non-convergence for RBPF was too high ($\sim$ 70%). Thereafter, RBPF will designate an improved version where a regularization step was added. The number of non-convergence is smaller for A2BC filters that are more robust than the benchmark filters.

Filters	% of non-convergence
RPF	8
A2BC-RPF	3
RBPF	10
A2BC-RBPF	6

TABLE I: Table of percentage rate of non-convergence for 100 Monte Carlo trials.

Scenario parameters

- Sampling period: $\Delta T = 10$ seconds
- Number of bathymetric measurements: 200
- Time at the end of the trajectory: $T = 33$ minutes
- Number of beams: 3
- Number of particles: $N = 5000$
- Resampling threshold: $N_{th} = 0.75N$
- ϵ domain (13): $I = [0.1, 20]$ meters
- Standard deviation of each beam range: $\sigma_R = 10$ meters

The initial uncertainty for the position P is $\sigma_P = diag([1000, 1000, 100])$ meters and the initial uncertainty for the velocity V is $\sigma_V = diag([0.5, 0.5, 0.5])$ meters per second. The initial state is $X_0 = [120000, 180000, -100, 5, 5, 0.05]^T$. The horizontal velocity vector is 7 meters per second. For the regularization step, we chose an Epanechnikov kernel. The filter results are compared for the same number of particles N.

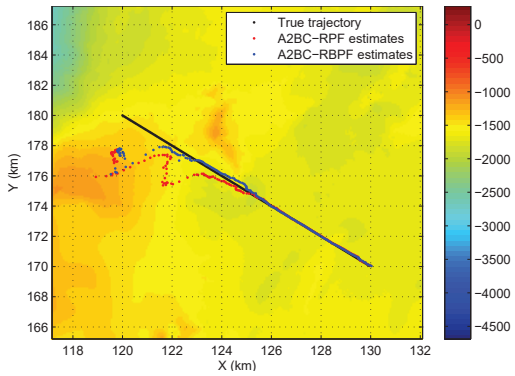


Fig. 4: Comparison between the true trajectory and the tajectories estimated by A2BC-RPF and A2BC-RBPF on the map of the California coast. The colorbar represents the depth levels in meters.

The reference trajectory is located in an ambiguous area of the map. Figure 4 illustrates the trajectories estimated by the two proposed algorithms in the same conditions (the same initial errors and measurements realizations). The filters converge quickly towards the reference trajectory.

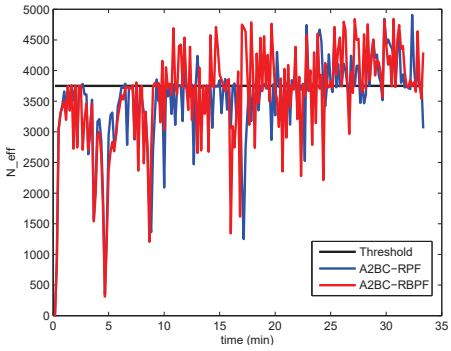
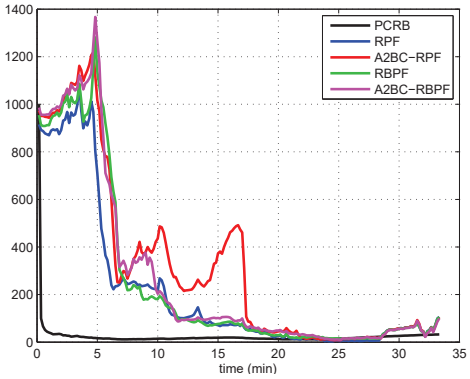
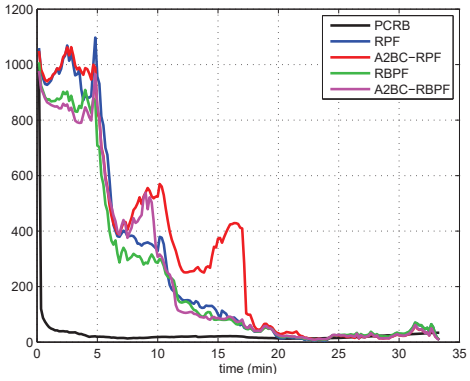


Fig. 5: Values taken by N_{eff} over time and resampling threshold N_{th}.

The criterion N_{eff} of the A2BC filters is plotted in Figure 5 with the resampling threshold N_{th}. The criterion varies over time to avoid the resampling step and therefore maintains the modes of the posterior density as long as possible (i.e. N_{eff} is forced to remain near N_{th}). It's the reason why in Figure 6, the A2BC-RPF has a higher error than the other filters between time 7 minutes and 17 minutes.



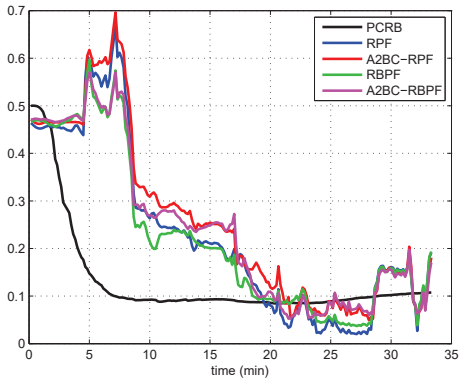
(a) RMSE of the position on the axis X.



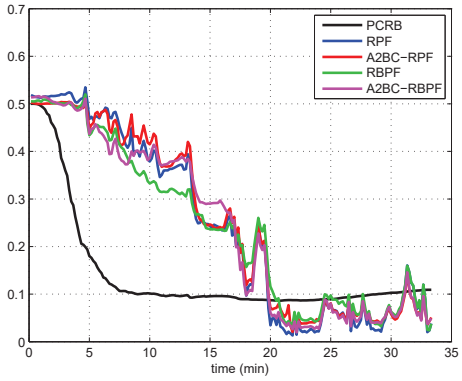
(b) RMSE of the position on the axis Y.

Fig. 6: PCRB and RMSE of the horizontal position in meters.

Figures 6 and 7 show the RMSE results of the four filters tested with similar simulation conditions. Only convergence cases are used to plot the curves on these two figures. Convergence generally occurs around 20 minutes. The curves follow the tendency of the approximate PCRB. The final accuracy of the filters is less than 100 meters in position, which corresponds to the spatial resolution of the map. The velocity accuracy is also satisfactory.



(a) RMSE of the velocity on the axis X.



(b) RMSE of the velocity on the axis Y.

Fig. 7: PCRB and RMSE of the horizontal velocity in meters per second.

The error at the end of the trajectory of the filters in horizontal position RMSE_T^{HP} and horizontal velocity RMSE_T^{HV} are reported in Table II. RMSE_T^{HP} is calculated as follows:

$$\text{RMSE}_T^{HP} = \sqrt{\text{RMSE}_T^{p_x\,2} + \text{RMSE}_T^{p_y\,2}} \tag{25}$$

as well for RMSE_T^{HV} by replacing $\text{RMSE}_T^{p_x}$ by $\text{RMSE}_T^{v_x}$ and $\text{RMSE}_T^{p_y}$ by $\text{RMSE}_T^{v_y}$. The results show that A2BC methods increase the accuracy in position and horizontal velocity.

Filters	RMSE_T^{HP}	RMSE_T^{pz}	RMSE_T^{HV}
RPF	106.53	1.47	0.195
A2BC-RPF	104.67	0.83	0.188
RBPF	105.23	1.46	0.197
A2BC-RBPF	99.90	0.78	0.177

TABLE II: Table of RMSE at the end of the trajectory (in meters for position and in meters per second for velocity).

V. CONCLUSIONS

In some cases, the observation model is a numerical black box, which leads us to consider an unknown likelihood. We introduce a measurement updating procedure based on an adaptive kernel derived from Approximate Bayesian Computational filters. The proposed method is implemented in two well-known particles filter: Regularized Particle Filter and Rao-Blackwellized Particle Filter. Simulation results demonstrate that the proposed method significantly increases the robustness and the accuracy of particle-like filters while remaining computationally efficient.

REFERENCES

[1] Yaakov Bar-Shalom, Peter K Willett, and Xin Tian. *Tracking and data fusion*. YBS publishing Storrs, CT, USA:, 2011.

[2] Pierre Del Moral, Jean Jacod, and Philip Protter. The monte-carlo method for filtering with discrete-time observations. *Probability Theory and Related Fields*, 120(3):346–368, 2001.

[3] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

[4] Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994.

[5] Tiancheng Li, Miodrag Bolic, and Petar M Djuric. Resampling methods for particle filtering: classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86, 2015.

[6] Jean-Michel Marin, Pierre Pudlo, Christian P Robert, and Robin J Ryder. Approximate bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.

[7] Deborah Kathleen Meduna. *Terrain relative navigation for sensor-limited systems with application to underwater vehicles*. Stanford University, 2011.

[8] José Melo and Aníbal Matos. Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Engineering*, 139:250–264, 2017.

[9] Christian Musso, Nadia Oudjane, and Francois Le Gland. Improving regularised particle filters. In *Sequential Monte Carlo methods in practice*, pages 247–271. Springer, 2001.

[10] Nadia Oudjane. *Stabilité et approximations particulaires en filtrage non linéaire application au pistage*. PhD thesis, Rennes 1, 2000.

[11] Dinh Tuan Pham. Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly weather review*, 129(5):1194–1207, 2001.

[12] Branko Ristic, Ajith Gunatilaka, Ralph Gailis, and Alex Skvortsov. Bayesian likelihood-free localisation of a biochemical source using multiple dispersion models. *Signal Processing*, 108:13–24, 2015.

[13] Thomas Schon, Fredrik Gustafsson, and P-J Nordlund. Marginalized particle filters for mixed linear/nonlinear state-space models. *IEEE Transactions on signal processing*, 53(7):2279–2289, 2005.

[14] B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26 of *Monographs on Statistics & Applied Probability*. Chapman and Hall, 1986.

[15] Petr Tichavsky, Carlos H Muravchik, and Arye Nehorai. Posterior cramér-rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on signal processing*, 46(5):1386–1396, 1998.