INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING
Int. J. Adapt. Control Signal Process. 2017; **31**:608–651
Published online 9 June 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/acs.2692

Robust particle filter formulations with application to terrain-aided navigation

Francisco Curado Teixeira[1,*,†], João Quintas[1], Pramod Maurya[2] and António Pascoal[1,2]

[1]*Institute for Systems and Robotics (ISR), Instituto Superior Técnico (IST), University of Lisbon, Portugal*
[2]*National Institute of Oceanography, Marine Instrumentation Division, Dona Paula, Goa, India*

SUMMARY

The work described in this paper is motivated by the need to develop efficient and robust estimation filters with application to terrain-aided navigation of underwater robotic vehicles. One of the main problems addressed is the development of navigation particle filters that can deal with the scarcity of landmarks and the terrain ambiguity that characterize vast regions of the ocean floor. As a contribution to solve this problem, the paper proposes three novel particle filter algorithms and assesses their estimation efficiency and robustness to non-informative measurements using two well-known benchmarking tests. The performance of the new filters in these tests demonstrates their potential to solve a class of nonlinear problems that include, but are not limited to, the type of underwater navigation problem that motivated the present work. Our study concludes by examining the performance of the filters in terms of determining the position and velocity of an autonomous underwater vehicle in the presence of unknown ocean currents. When applied to terrain-aided navigation, the novel particle filter formulations formulations mitigate filter divergence issues frequently caused by terrain symmetries and are more robust than other well-known versions when used in scenarios with poor terrain information. The theoretical developments presented and the results obtained in simulations are validated using real data acquired during tests with an autonomous marine robot.

Received 10 November 2014; Revised 26 March 2016; Accepted 20 April 2016

KEY WORDS: terrain-aided navigation; marine robotics; nonlinear filtering; Monte Carlo methods

List of Abbreviations

APF	auxiliary particle filter	PPF	prior-correction particle filter
ASV	autonomous surface vehicle	PF	particle filter
AUV	autonomous underwater vehicle	PMF	point mass filter
CRLB	Cramér-Rao lower bound	TAN	terrain-aided navigation
DVL	Doppler velocity logger	TERCOM	Terrain Contour Matching
GLPF	generic LPF	RMSE	root-mean-square error
GPS	global positioning system	RPF	regularized particle filter
INS	inertial navigation system	RAM	random-access memory
IS	importance sampling	SIR	sampling importance resampling
LPF	likelihood particle filter	SIS	sequential importance sampling
MPF	mixture particle filter	SISR	SIS + resampling
PDF	probability density function	SISR+R	SISR + roughening

*Correspondence to: Francisco Teixeira, Institute for Systems and Robotics (ISR), Instituto Superior Tcnico (IST), Universidade de Lisboa, Lisbon, Portugal.
†E-mail: fcurado@isr.ist.utl.pt

1. INTRODUCTION

1.1. Motivation and main contributions of the paper

Precise navigation is a fundamental requirement in the operation of underwater robotic vehicles for reasons of safety and to enable accurate georeferencing of the data acquired by these platforms. The by-now classic terrain-aided navigation (TAN) approach, originally demonstrated in air vehicles, offers a solution to the problem of autonomous vehicle navigation by relying on the observation of topographic features measured by a vehicle that are matched sequentially with a prior map of the terrain. In the context of underwater robotics, TAN has emerged in the last decade as a potentially affordable alternative to the conventional navigation of autonomous underwater vehicles (AUVs) based on inertial navigation systems (INS) and arrays of acoustic beacons. Because global positioning system (GPS) signals are not available underwater, long acoustic baselines often play a key role in the correction of the unbounded localization errors introduced by INS. However, the cost of high-grade inertial navigation systems and the deployment of artificial beacons is simply too high for a large number of scientific and commercial applications. The research effort dedicated to the development of TAN reliable systems is largely motivated by the need to develop a new class of affordable, multi-purpose AUVs capable of performing long-range and long-term navigation without incurring the costs of high-grade INS and the deployment of long baselines. Terrain navigation can be integrated with conventional dead-reckoning methods to achieve precise navigation in the short term combined with bounded localization errors in the long run. The cost of acquiring the reference maps required by TAN may become negligible if they are used repeatedly or shared by several users. Recent experiments in TAN have demonstrated the possibility of using reference maps, which are not available *a priori* but are acquired by the vehicle during the mission. It is thus expectable that future implementations of the terrain navigation concept can dispense with the need of using prior maps, by resorting to techniques similar to those used in simultaneous localization and mapping. An efficient implementation of this type of approach, accomplished using a Rao-Blackwellized particle filter, is described in [1] which includes a comprehensive list of fundamental references on simultaneous localization and mapping and TAN.

The potential of TAN has been demonstrated in the last few years by a series of experimental trials where sustained localization accuracy of a few meters was achieved during long periods of navigation over terrains of varying topographic richness. The experiments also demonstrated the feasibility of these methods using affordable navigation sensors. As consequence of this success, TAN is gaining increasing attention not only in the scientific community but also in the commercial sector.

Despite these recent advances, it is widely recognized that the terrain navigation approach is still in an experimental phase and that further research is necessary to transform TAN into a mature navigation methodology. At the core of the methodology are advanced algorithms for nonlinear filtering.

In the paper, we propose novel filtering algorithms included in the class of sequential Bayesian estimators usually designated as *particle filters* (PFs) and compare them with other well-known versions of this class of filters in terms of estimation efficiency and robustness. The development of new particle filters is motivated by the need to solve a number of estimation problems that arise in the context of TAN implementations. The issues considered include the estimation ambiguity caused by terrain symmetries and the occurrence of measurement artifacts that can be erroneously interpreted as terrain features. Both issues are often responsible for the induction of significant localization errors.

Because the convergence and stability properties of sample-based estimators such as particle filters in many practical applications of interest cannot be demonstrated analytically, it is fundamental to assess their performance through practical experiments, as far as possible in an exhaustive manner. It is extremely important, in particular, to analyze the asymptotic performance of these methods as their properties are justified theoretically by the law of large numbers. To achieve these objectives, the present work includes a series of Monte Carlo simulations, which in the case of the TAN tests, benefit from the availability of real navigation data and terrain information.

DOI: 10.1002/acs

Given the widespread application of PF to stochastic estimation and signal processing problems, we considered it relevant to start this study by analyzing and benchmarking the performance of the new filters when used to solve some classical problems documented in the literature. The analysis of the benchmark results provided us with valuable conclusions and hints for improved implementation of the TAN filters.

The main contributions of the paper are

- A comprehensive survey of prior work and a summary of new advances in TAN of underwater vehicles — Section 2.1.
- Three novel particle filter formulations designed to mitigate the negative effects of terrain ambiguity in TAN applications, with potential application to a broader class of signal processing problems — Section 4.
- The comparative assessment of the asymptotic performance of all the particle filter versions considered in the paper, an analysis which to our best knowledge had not been presented before — Section 5.
- A new, formal method to define the support of uniform prior distributions of particles in TAN designed to maintain the area of the support and the number of particles inside conservative bounds — Section 4.4.
- The solution of the TAN problem using the particle filters introduced in the paper and their experimental validation in water trials with a marine robotic vehicle — Sections 6.2 and 6.3.

The organization of the paper is as follows: in the remainder of this section, we introduce the notation and the performance metrics used in the following sections. In Section 2, we introduce the TAN problem and a summary of prior work on TAN. Section 3 introduces basic concepts and open issues in sequential Bayesian estimation and Monte Carlo methods that justify the developments of the following sections. Section 4 motivates and describes our novel particle filter formulations. In Section 5, we assess the performance of the new filters, applying them to solve two classical estimation problems. The performance of the proposed particle filter versions as applied to solve the TAN problem is illustrated in Section 6. Finally, in Section 7, we summarize the results obtained and present the main conclusions of the paper.

1.2. General system models and notation

We start by introducing the key notation used in the paper. See, for example, [2, 3] and the references therein for some basic concepts on nonlinear filtering with special emphasis on PFs. In what follows, $\mathbf{x}$ is a vector representing a system state with dimension n_x, and $\mathbf{y}$ is a measurement vector with dimension n_y; N is the number of samples (particles) used by the particle filter; $\mathbf{x}_t^i$ represents the ith particle (a random sample from the state space) at time instant t, and ω_t^i denotes the weight associated to particle $\mathbf{x}_t^i$. Additionally, $\mathbf{X}_t$ represents the sequence of states $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_t$, and $\mathbf{Y}_t$ represents the sequence of observations $\mathbf{y}_1, ..., \mathbf{y}_t$. The nonlinear systems analyzed in the paper are represented in state-space form as

$$\begin{aligned} \mathbf{x}_{t+1} &= f(\mathbf{x}_t) + \mathbf{w}_t \\ \mathbf{y}_t &= h(\mathbf{x}_t) + \mathbf{v}_t \end{aligned} \tag{1}$$

where $\mathbf{w}_t \in \mathbb{R}^{n_w}$ and $\mathbf{v}_t \in \mathbb{R}^{n_v}$ are the process and measurement noises with probability density functions (PDFs) $f_w(\mathbf{x}_{t+1}|\mathbf{x}_t) \equiv p(\mathbf{x}_{t+1} - f(\mathbf{x}_t)) = p(\mathbf{w}_t)$ and $h_v(\mathbf{y}_t|\mathbf{x}_t) \equiv p(\mathbf{y}_t - h(\mathbf{x}_t)) = p(\mathbf{v}_t)$, respectively, which we assume are independent and known for every time t; n_w and n_v are the dimensions of the noise vectors. In the context of terrain navigation, $h(.) : \mathbb{R}^2 \to \mathbb{R}$ is typically an elevation map, a nonlinear, non-structured function. For the sake of simplicity of notation, we use the notation $p_T(.)$ to represent the PDF history $p_{0:t}(.)$.

1.2.1. Main assumptions. The state of the system, $\mathbf{x}_t \in \mathbb{R}^{n_x}$ is considered to be a Markov process, that is,

DOI: 10.1002/acs

$$p\left(\mathbf{X}_t\right) = p\left(\mathbf{x}_0\right) \prod_{k=1}^{t} p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right) \tag{2}$$

with $p(\mathbf{x}_0)$ denoting the prior distribution for the state at $t = 0$. The observations are conditionally independent given the states, that is,

$$p\left(\mathbf{Y}_t|\mathbf{X}_t\right) = \prod_{k=1}^{t} p\left(\mathbf{y}_k|\mathbf{x}_k\right). \tag{3}$$

1.2.2. Performance evaluation metrics. The metrics used in the paper for filter performance evaluation are defined as follows. Let M be the number of independent Monte Carlo (MC) simulations (runs) and let $t = 0, ..., T$ be the time span of each simulation. To facilitate quantitative comparisons with the results reported in the literature, we use the traditional measure of performance: the root-mean-squared error (RMSE). In the case of a one-dimensional problem, we use two distinct metrics: the time-indexed RMSE, which is computed over the ensemble of MC runs according to the expression

$$RMSE_t = \sqrt{\frac{1}{M}\sum_{k=1}^{M}(\hat{\mathbf{x}}_{t_k} - \mathbf{x}_{t_k})^2}$$

and the MC ensemble average of the RMSE computed over the time span $t = 0, ..., T$, that is,

$$RMSE^M = \frac{1}{M}\sum_{k=1}^{M}\sqrt{\frac{1}{T}\sum_{t=0}^{T}(\hat{\mathbf{x}}_{t_k} - \mathbf{x}_{t_k})^2}$$

where $\mathbf{x}_{t_k}$ and $\hat{\mathbf{x}}_{t_k}$ denote the true and the estimated state at time t of the kth MC run, respectively.

For two-dimensional position estimation, including the TAN problem, we present the time-indexed RMSE of position defined by

$$RMSE_t^M = \sqrt{\frac{1}{M}\sum_{k=1}^{M}(\hat{x}_{t_k} - x_{t_k})^2 + (\hat{y}_{t_k} - y_{t_k})^2}$$

where coordinates (x_{t_k}, y_{t_k}) and $(\hat{x}_{t_k}, \hat{y}_{t_k})$ denote the true and the estimated target positions at time t of the kth MC run, respectively.

2. THE TERRAIN-AIDED NAVIGATION PROBLEM

The techniques of geophysical-based navigation of autonomous underwater vehicles rely on the solution of nonlinear estimation problems that address explicitly the availability of geophysical data extracted from the terrain (e.g., topographic and geomagnetic data). In the case where only topographic (bathymetric) data are used, the resulting navigation systems fall in the scope of what is commonly called terrain-aided navigation. In this case, and in its simplest form, local bathymetry information is obtained by a marine vehicle using an echo sounder and a depth cell that provide measurements of altitude above the seabed and depth, respectively. More advanced systems of this type employ multiple, simultaneously fired acoustic beams to acquire the terrain measurements. The data obtained are then compared with an existing bathymetric map of the area, stored on-board the underwater vehicle. Filtering techniques that merge this information with that obtained with dead reckoning data are then used to estimate the vehicle's position and velocity on-line.

DOI: 10.1002/acs

2.1. Prior work on terrain-aided navigation

The application of TAN techniques to underwater robotic vehicles has been, to a great extent, influenced by the successful developments of TAN for military aerial vehicles. Well-known examples of these systems are the early Terrain Contour Matching (TERCOM) method developed in the 1950s (e.g., [4]) and Sandia Inertial Terrain-Aided Navigation, a terrain-referenced navigation algorithm for helicopters described in [5].

The earliest TAN algorithms implemented some form of correlation of profiles or terrain patches to estimate the position of a vehicle. More recent implementations apply sequential estimation methods based on single-point measurements and point-mass approximations of probability distributions (e.g., [6]).

2.1.1. Recent advances in terrain-aided underwater navigation. To the best of our knowledge, prior to 2004 there were only a few references mentioning the utilization of terrain navigation by military submarines of the US Navy (e.g., [7] and [8]) and some publications on the development of prototype systems for autonomous underwater vehicles. Examples of the latter are [9] and [10].

A survey of terrain-based navigation methods and systems up to 2010 is presented in [11]. A more recent review of the state of art on TAN can be found in [12]. We present later a summary of the developments reported in the literature, documenting the evolution of TAN during the last 10 years. The review focuses on work that illustrates terrain-navigation efficacy through experimental trials and the use of real data.

In [13], the authors describe the adaptation of the TERCOM algorithm to exploit single and multiple-ping bathymetric profiles acquired along-path by a multibeam echo-sounder installed on a *HUGIN* class AUV of the Norwegian Defence Research Establishment; the efficacy of the solution is demonstrated in simulations using real AUV navigation data and high-resolution bathymetric maps.

The terrain navigation module described in [14] exploits multi-beam and synthetic aperture sonar data and is integrated with the INS of the SAAB *AUV62F* and *Sapphires* AUVs operated by the Swedish Defense for surveillance and mine reconnaissance. A navigation system incorporated in the Autosub 6000 AUV that uses multi-beam sonar data to correct the combined inertial-Doppler (INS-DVL) dead-reckoning navigation drift is presented in [15].

The first report of a successful low-cost terrain navigation solution relying on altitude data supplied by a four-beam DVL is described in [16]; the follow-up of the work, including tests with the MBARI *Dorado*, is documented in [17] and [18]. A similar TAN implementation based on altitude data supplied by a Doppler velocity logger was developed for the *HUGIN* AUV family as reported in [19] and [20]. A commercial version of this system has been released recently ([21]).

For navigation at low altitude (up to 20 m) where downward-looking sonar beams provide a too-narrow swath coverage of the terrain that can be insufficient for terrain navigation, a combination of multi-beam and interferometric synthetic aperture sonar, which has a wide swath even at low altitudes, is reported in [22].

Based on extensive datasets collected with the *MARV* AUV, [23] discusses the main causes of terrain-navigation systems failures due to a combination of technological and algorithmic issues and proposes practical solutions to these problems.

2.1.2. Particle filters applied to terrain-aided navigation. The application of sequential Monte Carlo estimation methods and their superior performance relative to parametric estimation approaches (e.g., Kalman-based filters) and matching processes such as the TERCOM are reported among others in [10, 15, 16, 24–30], and [18]. The computation of lower bounds of unbiased and biased estimators in terrain-based navigation is addressed in [31] as well as in some of the previously cited publications.

One of the main problems posed by the application of PFs is filter divergence due to the *degeneracy* phenomenon. The issue is well documented in the signal processing and estimation literature, and has received considerable attention by the TAN community. In [32], a local regularized rejection PF is introduced, which implements a novel *regularization* method at the prediction step of the

DOI: 10.1002/acs

filter based on the regularized particle filter (RPF) proposed in prior theoretical work by the same authors. The application of L2RPF to tracking applications, including the bearings-only problem, is documented in [33]. A comparative performance assessment of regularized classical particle filters is presented in [34]. Despite the improvements achieved by regularized PFs in terms of overall estimation accuracy, the filters have been shown to achieve reduced converge speeds because of the addition of artificial noise to particle trajectories in the regularization step (e.g., [35]).

The susceptibility of navigation filters to terrain symmetries is one of the main open problems in the context of terrain navigation. Terrain ambiguity induces multi-modal probability distributions of position that eventually lead to filter divergence in TAN. To mitigate this problem from an algorithmic standpoint, a new particle filter version designated the smooth kernel particle filter is introduced in [36]. With a similar objective but taking a different approach, [14] proposed the utilization of maximum *a posteriori* likelihood, instead of minimum mean square error estimates. In [37], it was demonstrated that the resampling operation of particle filters automatically leads to loss of multi-modality in finite time, eventually leading to filter divergence in contexts of ambiguous measurements. The solution proposed to overcome this problem builds on the aforementioned work [32] and utilizes a mixture implementation of the RPF where each mixture component corresponds to a mode of the filtering distribution.

A topic that is deserving increased attention in the literature on TAN is the implementation of techniques for detection of false convergence of terrain navigation filters. Well-known examples of adverse conditions that can lead to TAN filter divergence include operating over flat terrains or areas of time-changing topography, the utilization of reference maps with artifacts resulting from the map-generation process, and faulty sensor readings. The problem is addressed in [38] and [39] through the inclusion of false convergence detectors in the navigation PF. The recent work [40] addresses this subject explicitly through the application of a robust navigation algorithm that mitigates the effects of outliers in terrain measurements and in the terrain model (map). Other potential causes of failure in terrain navigation, including bias in orientation and attitude sensors and inaccuracies in the reference map, are mentioned and proposed as a subject of future research in [18]. In [41], the authors analyze the most common causes of filter divergence in terrain navigation and propose a framework for fault detection and filter recovery in TAN systems. The efficacy of the method is illustrated in offline simulations using field data acquired with the MBARI *Dorado* AUV.

The design and experimental tests of a PF-based robust navigation system with fault diagnosis for a remotely operated underwater vehicle are described in [42]. Given its ability to diagnose failures in Doppler-based systems, including velocity biases and DVL dropouts, the method has high potential of application to terrain-based navigation methods that rely on DVL for simultaneous dead-reckoning and terrain data acquisition.

In the PF algorithms that we present, the detection of filter divergence is implemented based on the sum of the weights of the particles, Σ_w. An extremely low value of Σ_w, which holds for several iterations, is an indicator that the filter has diverged. In this situation, which is similar to the *kidnapped robot problem*, a well-known solution is the re-initialization of the filter by distributing the particles uniformly on the whole map. The experimental results show that the procedure normally succeeds if, a few iterations after re-initialization, the vehicle reaches a position where the terrain provides unambiguous information. This situation is illustrated by a practical experiment in paragraph 6.3.2.

2.2. Terrain-aided navigation problem formulation and notation

The following notation is used in the TAN problem formulation and in the kinematic equations that will be introduced in Section 6: $\{I\}$ represents an inertial coordinate frame, $\{B\}$ denotes the body-fixed frame that moves with the vehicle, $\mathbf{p} = [x, y, z]^\top$ is the position of the origin of $\{B\}$ measured in $\{I\}$, $\lambda = [\phi, \theta, \psi]^\top$ represents roll, pitch and yaw angles that parameterize locally the orientation of $\{B\}$ relative to $\{I\}$, and $\omega = [p, q, r]^\top$ represents the angular velocity of $\{B\}$ w.r.t. $\{I\}$, expressed in $\{B\}$.

DOI: 10.1002/acs

2.2.1. Terrain-aided navigation problem formulation. The TAN problem formulation presented here is motivated essentially by the need to develop state estimators for underwater vehicle navigation when a reliable dynamic model of the vehicle is not available. For navigation purposes, it may be appropriate to rely on the vehicle kinematics only, for the relevant input variables that consist of body-axis linear and rotational velocities can be measured directly using Doppler and attitude and heading reference systems, respectively. This dispenses with the need to measure accelerations, either directly using expensive sensor suites, or indirectly by using a model of the vehicle and measuring the force and torque imparted to it by the vehicle actuators (e.g., propellers, deflection surfaces, etc.). Although the inclusion of a dynamic model of the vehicle may bring some form of robustness against temporarily missing kinematic sensor data, the price to be paid (both in terms of complexity of the navigation algorithms that must be implemented, cost associated with the identification of a good enough dynamic model of the vehicle, and the estimation of dynamics-related data such as forces and torques produced by actuators) may render a more sophisticated navigation system unacceptable for small, cost-effective AUVs. The problem is formulated as follows:

Given the kinematic model of an underwater vehicle, an observation model of the characteristics of the terrain that are measured sequentially, a previously acquired map of the terrain variables of interest, and the velocity of the vehicle with respect to the water expressed in $\{B\}$, estimate the position and the velocity of the vehicle relative to $\{I\}$ in the presence of constant or slowly varying unknown ocean currents.

2.2.2. Terrain-aided navigation system setup. To solve the TAN problem formulated in the preceding text, the present study proposes navigations algorithms for AUVs that exploit real topographic information acquired by narrow-beam echo-sounders to complement the estimation based on the kinematic model of the vehicle and compensate for the velocity bias introduced by unknown ocean currents. The performance of these navigation methods is analyzed in simulated trials using a real map of the terrain and a sensor configuration that mimics the real system used to obtain the bathymetry. The data were acquired by an autonomous surface vehicle (ASV) that was not able to measure its speed-over-ground using sonar sensors. The ASV localization in real time was ensured by GPS; these data are used here as ground-truthing to validate the TAN position estimates. The navigation algorithms developed may also be applied to a vehicle that navigates close to the sea-bottom and has the ability to measure its ground-referenced velocity using for example a DVL working in bottom-lock. In this case, the filter set-up adopted can be used as exposed in [43] to estimate and compensate the inherent drift of the dead-reckoning approach in order to guarantee bounded localization error. The detailed characteristics of the real sensors used in the current study are presented in Section 6.

3. SEQUENTIAL BAYESIAN ESTIMATION AND IMPORTANCE SAMPLING

3.1. Prior and related work

The algorithms of the different particle filter implementations discussed in the paper are presented in the subsequent sections in a high-level language (pseudo-code). Some details of the alternative PF implementations are not discussed here because they are widely covered in the referenced literature. The interested reader can find in, for example, [2, 44], and [45], comprehensive and in-depth expositions of these methods and extensive lists of references on the subject.

3.2. Sequential Bayesian estimation procedures

A sequential Bayesian estimation procedure based on the state-space model of a dynamical system with application to continuous and discrete density functions was proposed originally in [46]. We adopt its formulation and notation in the forthcoming discussion.

Under some assumptions compatible with the conditions of the problem we address presently, the posterior density $p_{T|T}(\mathbf{X}_t|\mathbf{Y}_t)$ can be evaluated following a recursive procedure consisting of two stages:

DOI: 10.1002/acs

Prediction

$$p_{T|T-1}(\mathbf{X}_t|\mathbf{Y}_{t-1}) = p_{T-1|T-1}(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1})\, f_w(\mathbf{x}_t|\mathbf{x}_{t-1})$$

Correction

$$\begin{aligned} p_{T|T}(\mathbf{X}_t|\mathbf{Y}_t) &= \frac{h_v(\mathbf{y}_t|\mathbf{x}_t)\, p_{T|T-1}(\mathbf{X}_t|\mathbf{Y}_{t-1})}{l_{T|T-1}(\mathbf{y}_t|\mathbf{Y}_{t-1})} \\ &= \frac{h_v(\mathbf{y}_t|\mathbf{x}_t)\, p_{T|T-1}(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_{t-1})\, p_{T-1|T-1}(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1})}{l_{T|T-1}(\mathbf{y}_t|\mathbf{Y}_{t-1})} \end{aligned}$$

where, for a fixed observation $\mathbf{y}_t$, the scaling factor $l_{T|T-1}(\mathbf{y}_t|\mathbf{Y}_{t-1})$ becomes a normalizing constant. The former is a conceptual solution of limited applicability, however. In general, there is no closed-form expression for the conditional density $p_{T|T}(\mathbf{X}_t|\mathbf{Y}_t)$ due mainly to the unstructured nonlinearity of $h_v(\mathbf{y}_t|\mathbf{Y}_t)$; as a consequence, this conditional density cannot be updated analytically. For example, in the case of linear state and measurement models affected by Gaussian noise, $p_{T|T}(\mathbf{X}_t|\mathbf{Y}_t)$ can be evaluated analytically and the optimum MMSE estimator is the Kalman filter. In the case of TAN estimators, $h_v(.)$ is typically a non-structured, nonlinear map, and there is no closed-form expression for the likelihood function $h_v(\mathbf{y}_t|\mathbf{Y}_t)$.

Because in general it is impossible to sample directly from $p_{T|T}(\mathbf{X}_t|\mathbf{Y}_t)$, an importance sampling procedure may be adopted to sample from an approximation of the posterior designated *importance distribution* (e.g., [47]). For this purpose, assume that it is easy to draw a set of N sample paths $\mathbf{X}_t^i\,(i = 1, ..., N)$ from a convenient importance (proposal) distribution $q_T(\mathbf{X}_t|\mathbf{Y}_t)$. According to the importance sampling principle, a set of non-normalized importance weights may be computed as

$$\tilde{w}_T^i = \frac{p_{T|T}(\mathbf{X}_t^i|\mathbf{Y}_t)}{q_T(\mathbf{X}_t^i|\mathbf{Y}_t)}\,,\ (i = 1, ..., N). \tag{4}$$

The weighted sample $\{(\mathbf{X}_t^i, \tilde{w}_t^i)\}_{1\leqslant i\leqslant N}$ so obtained may be used to approximate the expectation of any function $g(.)$ defined on the state space:

$$\hat{g} \equiv E_{p_{T|T}}\{g(\mathbf{X}_t|\mathbf{Y}_t)\} = \int g(\mathbf{X}_t)\, p_{T|T}(\mathbf{X}_t|\mathbf{Y}_t) d\mathbf{X}_t \approx \sum_{i=1}^{N} w_t^i\, g\left(\mathbf{X}_t^i\right) \tag{5}$$

where $w_t^i = \tilde{w}_t^i \Big/ \sum_{j=1}^{N} \tilde{w}_t^j$ is the normalized weight at time t over path i and $\mathbf{X}_t^i$ represents the ith sample path of variable $\mathbf{x}$ [48].

In order to simplify the expressions used in the evaluation of the weights, the proposal distribution may be factorized as

$$q_T(\mathbf{X}_t|\mathbf{Y}_t) = q_{T-1}(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1}) q_T(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t).^{\ddagger}$$

Using the property of conditional independence of the observations given the states, that is,

$$q_T(\mathbf{Y}_t|\mathbf{X}_t) = q_0(\mathbf{x}_0) \prod_{k=1}^{t} q_k(\mathbf{y}_k|\mathbf{x}_k)$$

and applying Bayes' and chain rules, it is straightforward to show that

$$q_T(\mathbf{x}_t|\mathbf{X}_{t-1}, \mathbf{Y}_t) \propto q_T(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t).$$

‡Although this equality is normally used in the literature, a proportionality sign should be used instead of the equality sign. However, it can be easily shown that the correction factor, which is omitted in the equation, is a constant that does not affect the evaluation of the importance weights.

DOI: 10.1002/acs

As such, it is possible to discard the path $\mathbf{X}_{t-1}^i$ and the history of the observations $\mathbf{Y}_{t-1}$ in the probability density function $q_T(.)$ proposed to model the process error. As a consequence, the importance weight update equation takes the form

$$\tilde{w_t^i} = \frac{p_{T|T}\left(\mathbf{X}_t^i|\mathbf{Y}_t\right)}{q_T\left(\mathbf{X}_t^i|\mathbf{Y}_t\right)} = \frac{h_v\left(\mathbf{y}_t|\mathbf{x}_t^i\right) p_{T|T-1}\left(\mathbf{x}_t^i|\mathbf{X}_{t-1}^i, \mathbf{Y}_{t-1}\right) p_{T-1|T-1}\left(\mathbf{X}_{t-1}^i|\mathbf{Y}_{t-1}\right)}{q_T\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_t\right) q_T\left(\mathbf{X}_{t-1}^i|\mathbf{Y}_{t-1}\right)}$$

$$= \tilde{w}_{t-1}^i \frac{p_{T|T-1}\left(\mathbf{x}_t^i|\mathbf{X}_{t-1}^i, \mathbf{Y}_{t-1}\right) h_v\left(\mathbf{y}_t|\mathbf{x}_t^i\right)}{q_T\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_t\right) l_{T|T-1}\left(\mathbf{y}_t|\mathbf{Y}_{t-1}\right)}.$$

Noting that $l_{T|T-1}(\mathbf{y}_t|\mathbf{Y}_{t-1})$ is a constant, independent of $\mathbf{x}_t^i$, and using the Markov property of the process model, we obtain the recursive formula for weight updating:

$$\tilde{w}_t^i = w_{t-1}^i \frac{f_w\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i\right) h_v\left(\mathbf{y}_t|\mathbf{x}_t^i\right)}{q_T\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_t\right)}. \tag{6}$$

This recursive expression can be found at the core of most sequential Bayesian estimation procedures commonly designated as PFs.

3.3. Importance sampling and particle filter implementation issues

Importance sampling (IS) is a biasing method designed to reduce the sampling variance in Monte Carlo simulations, fundamental to the developments presented in this paper. In-depth expositions of this topic may be found in the following books [47, 49–51]. One of the main issues inherent to IS-based algorithms such as particle filters is the *degeneracy* phenomenon. This and the related problems of *sample impoverishment* and *particle collapse* are given particular attention in the presented study. The interested reader may find these topics documented, for example, in [2]. In the paper, we refer several *ad hoc* techniques that have been proposed in the literature to solve the problem of sample impoverishment. Among the most frequently mentioned, we consider *prior boosting*, *roughening*, *prior editing*, and *regularization*, which are described in [2, 52], and [33].

3.4. Importance sampling and choice of importance density

3.4.1. Choice of importance density. In practical implementations, it is necessary to choose a convenient importance density q_t from which the set of samples will be obtained at the prediction stage of the filter. Actually, the choice of this sampling density leads to different filter versions and is crucial for the performance of the particle filter. A thorough discussion of the impact of the importance densities on filter performance can be found in [44, 51]. It is well known that the optimal sampling density corresponds to the posterior, $p_{T|T-1}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y})$, which is the actual underlying density governing the evolution of the process. In this case, the value of the weight computed for each particle is independent of the particle sampled, and the variance of the weights is zero [2]. All the alternative proposal densities increase the variance of the weights, thus reducing the effective number of particles. This topic is analyzed in more detail in Section 4.4. It is well known, however, that using the optimal importance density is not feasible except in some special cases where the posterior is a density function that can be represented or approximated analytically with great accuracy (e.g., [2, 53, 54]). A simplified implementation introduced in the sampling importance resampling filter (SIR) or *bootstrap* algorithm, as presented in [52], utilizes the prior $p_{T|T-1}(\mathbf{x}_t|\mathbf{x}_{t-1}) = f_w(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$ as the proposal density. This choice of importance density is very convenient because it leads to a simple and intuitive expression for the correction stage where the new samples drawn from the prior have their weights updated by the likelihood

$$w_t^i \propto w_{t-1}^i h_v\left(\mathbf{y}_t|\mathbf{x}_t^i\right). \tag{7}$$

This approach is also applied in the sequential importance sampling (SISR) filter, which is derived from the SIR (Algorithm 1). Given its widespread utilization as the core algorithm of particle

DOI: 10.1002/acs

filter implementations, we designate the SISR as the *standard PF* algorithm. Although (7) constitutes a practical formula used by several filter implementations, the proposal density that leads to this expression presents some disadvantages. The resultant particles do not have minimum-variance weights, which in practice corresponds to reducing the effective number of samples; consult paragraphs 4.4.1 and 5.1.2 for an introduction to this topic. An additional issue regarding the choice of the importance distribution is related to a basic requirement of the importance sampling principle: IS assumes that the support of the importance distribution includes the support of the target density [47]. This may not be verified when the importance density is too peaked or too skewed relatively to the filtering density.

The novel PF versions that are described later and embodied in Algorithms 3, 4, and 6 address these problems explicitly through the use of alternative importance densities. However, the developments introduced are motivated mainly by the need to choose and adequate *filtering density*, which is demonstrated to be even more relevant in TAN applications. This subject is discussed in the following section. For convenience of the reader, we present here the conventional SISR algorithm as proposed in [52], and the auxiliary particle filter (APF) and the likelihood particle filter (LPF) according to the formulations presented in [55]; see Algorithms 1, 2, and 5, respectively. In the algorithms that do not execute systematic resampling, the parameter K_{eff} is multiplied by the sample size to obtain the threshold on the effective number of particles that conditions the execution of the resampling step. The variable S_u represents the subset of the state space used to generate the uniformly distributed particles, when applicable.

Algorithm1 : SISR

$\left[\left\{\mathrm{x}_k^i, w_k^i\right\}_{i=1}^N\right] = \text{SISR}\left[\left\{\mathrm{x}_{k-1}^i, w_{k-1}^i\right\}_{i=1}^N, \mathrm{y}_k, N\right]$

01. FOR $i = 1 : N$
02. $Draw \quad \mathrm{x}_k^i \sim p(\mathrm{x}_k^i|\mathrm{x}_{k-1}^i)$
03. $w_k^i = w_{k-1}^i p(\mathrm{y}_k|\mathrm{x}_k^i)$
04. END FOR
05. S = SUM $\left[\left\{w_k^i\right\}_{i=1}^N\right]$
06. FOR $i = 1 : N$
07. $w_k^i = w_k^i \cdot S^{-1}$
08. END FOR
09. $\left\{x_k^i, w_k^i, -\right\}_{i=1}^N = \text{RESAMPLE}\left[\left\{\mathrm{x}_k^j, w_k^j\right\}_{j=1}^N\right]$

Alternative implementations of the RESAMPLE procedure used in Algorithms 2 to 5 can be found in [2]. In practical implementations, we apply *residual resampling* due to the reasons explained in [36]. Given the impact of resampling techniques on filter performance, this topic has received considerable attention from the statistics community and is well documented in the literature (e.g., [56, 57]). The pseudo-code for other filter implementations can be found in [2].

4. NOVEL FORMULATIONS OF THE PARTICLE FILTER

4.1. Motivation-preventing filter divergence in terrain-aided navigation applications

The results of computer simulations obtained in our previous work [36] put in evidence the deficient robustness of the standard particle filter in the presence of small to medium-scale symmetries[§] in the terrain topography. Given its ability to represent multi-modal distributions, the filter diverges

[§]We do not employ a metric to establish the scale of the topographic symmetries. This scale can be related to the size of the area of spreading of the particles in successive filter iterations: a small-scale topographic symmetry is enclosed by the support of the approximate probability distribution of the vehicle localization, thus leading to multi-peaked likelihoods; a large-scale terrain symmetry is not covered by that support and is not susceptible of causing ambiguity in

DOI: 10.1002/acs

Algorithm2 : **APF**

$\left[\left\{\mathrm{x}_k^i, w_k^i\right\}_{i=1}^N\right] = \text{APF}\left[\left\{\mathrm{x}_{k-1}^i, w_{k-1}^i\right\}_{i=1}^N, \mathrm{y}_k, N\right]$

FOR $i = 1 : N$
  $Draw \quad \mu_k^i \sim p(\mathrm{x}_k|\,\mathrm{x}_{k-1}^i)$
  $\nu_{k-1}^i = p(\mathrm{y}_k|\mu_k^i) \cdot w_{k-1}^i$
END FOR
$\left\{-, -, i^j\right\}_{i=1}^N = \text{RESAMPLE}\left[\left\{\mathrm{x}_{k-1}^j, \nu_{k-1}^i\right\}_{j=1}^N\right]$
FOR $j = 1 : N$
  $Draw \quad \mathrm{x}_k^i \sim p(\mathrm{x}_k|\,\mathrm{x}_{k-1}^i)$
  $w_k^j = \frac{w_{k-1}^{i^j}}{\nu_{k-1}^{i^j}} p(\mathrm{y}_k|\,\mathrm{x}_k^i) = \frac{p(\mathrm{y}_k|\,\mathrm{x}_k^i)}{p(\mathrm{y}_k|\mu_k^{i^j})}$
END FOR
S = SUM $\left[\left\{w_k^i\right\}_{i=1}^N\right]$
FOR $i = 1 : N$
  $w_k^i = w_k^i \cdot S^{-1}$
END FOR

Algorithm3 : **MPF**

$\left[\left\{\mathrm{x}_k^i, w_k^i\right\}_{i=1}^N\right] = \text{MPF}\left[\left\{\mathrm{x}_{k-1}^i, w_{k-1}^i\right\}_{i=1}^N, \mathrm{y}_k, N_s, N_e, K_{eff}, S_u\right]$

$N = N_s + N_e$
$N_T = N \cdot K_{eff}$
FOR $i = 1 : N_s$
  $Draw \quad \mathrm{x}_k^i \sim p(\mathrm{x}_k^i|\mathrm{x}_{k-1}^i)$
  $w_k^i = w_{k-1}^i p(\mathrm{y}_k|\mathrm{x}_k^i)$
END FOR
FOR $i = N_s + 1 : N$
  $Draw \quad \mathrm{x}_k^i \sim Uniform(S_u)$
  $w_k^i = w_{k-1}^i p(\mathrm{y}_k|\mathrm{x}_k^i) p(\mathrm{x}_k^i|\mathrm{x}_{k-1}^i)$
END FOR
S = SUM $\left[\left\{w_k^i\right\}_{i=1}^N\right]$
FOR $i = 1 : N$
  $w_k^i = w_k^i \cdot S^{-1}$
END FOR
$\hat{N}_{\text{eff}} = 1/\text{SUM}\left[\left\{\left(w_k^i\right)^2\right\}_{i=1}^N\right]$
IF $\hat{N}_{\text{eff}} < N_T$
  $\left\{x_k^i, w_k^i, -\right\}_{i=1}^N = \text{RESAMPLE}\left[\left\{\mathrm{x}_k^j, w_k^j\right\}_{j=1}^N\right]$
END IF

if the wrong modes are favored in a sequence of iterations sufficiently large to make the particles associated to the true mode collapse. This phenomenon is justified in part by the practical implementation of the standard PF. Frequently, the transition kernel that represents the stochastic model of system dynamics needs to be smoothed, as illustrated in [39], to avoid filter divergence due to the poor overlap of the prior and the likelihood. In practice, some of the techniques mentioned in Section 3.3 to address the problem of sample impoverishment correspond to *ad hoc* or formal

the estimation; medium-scale symmetries may become enclosed by the support of the distribution when it is enlarged adaptively in consecutive iterations due to the poor likelihood of the observations.

DOI: 10.1002/acs

methods of kernel smoothing. Drawing samples from a smoothed prior generates *process-model outliers* (improbable states according to the true, non-smoothed process model) that may correspond to high observation likelihoods due to the terrain ambiguity. These samples will have their weights amplified at the update stage of the filter. A similar issue is posed by the occurrence of measurement artifacts or outliers in the terrain map that can be interpreted as real features and have large likelihoods (e.g., [40]). This scenario is likely to occur in the standard PF implementation because the update stage of the filter is based only on the likelihood of measurements. The acceptance of the outliers that have their weights greatly amplified relatively to process model plausible states generates more diffuse posteriors and biased estimates. The propagation of these errors along time is a common cause of filter divergence, especially when the representation of the system probability densities relies on low-dimensional marginals such as the predictive distribution $p_{t+1|t}$ and the filtering distribution $p_{t|t}$ instead of the respective PDF histories $p_{T+1|T}$ and $p_{T|T}$.

Besides our experiments with the SISR particle filter in prior work, we tested alternative PF algorithms, including the auxiliary particle filter and the regularized particle filter. The main difficulties found in the implementation of the RPF and the plausible causes of the poor results obtained with the APF are discussed in [36]. The poor performance of the APF in the context of terrain-based navigation had already been observed in [58]. Given the poor results obtained with those PF implementations, our research effort aimed at the development of a new particle filter version that has shown improved convergence properties and robustness to outliers (e.g., [36, 59]).

In this section, we introduce three novel particle filter algorithms, the mixture particle filter (MPF), the prior-correction particle filter (PPF), and the generic likelihood particle filter (GLPF). A new method for the implementation of non-informative prior distributions in bi-dimensional estimation problems is also proposed; this method is integrated in the PF algorithms used in Section 6 to solve the TAN problem. The proposed algorithms are benchmarked in the following sections.

4.2. Mixture particle filters

Global localization failures, which often occur in the presence of terrain symmetries, are a consequence of the sample impoverishment introduced by the resampling procedure employed to mitigate the degeneracy phenomenon. To solve the problem, in [60], the authors propose the inclusion of uniformly distributed particles in the weighted sample used to compute the expectation of the vehicle kinematic state. One of the advantages of the method, if correctly parameterized, is to ensure that the proposal has a sufficiently large support to include that of the posterior.

Although lacking a theoretical support for the derivation of the importance sampling weights adapted to the resultant mixture distribution, the inclusion of additional particles proves to be very efficient at solving the problem of sample impoverishment. In simulations of TAN, it effectively avoids the sample collapse that makes the filter diverge in common terrain scenarios. An undesirable effect of the method is the generation of process-model outliers when samples are drawn from the uniform distribution. If not filtered adequately, these outliers can lead to filter divergence due to symmetries in the terrain.

To explore this circle of ideas, a modified proposal distribution is introduced in [60, 61] whereby a subset of the particles is propagated according to the PDF of the stochastic error model $f_w(\mathbf{x}_t|\mathbf{x}_{t-1})$ as in a standard implementation; another subset corresponding to a small fraction of particles (e.g., 5%) is generated by the measurement likelihood $h_v(\mathbf{y}_t|\mathbf{x}_{t-1})$. The main objective of this approach is to generate a subset of particles with large measurement likelihoods in order to avoid particle degeneracy. However, the implementation proposed is only possible if the inverse of the measurement model has a closed form solution from which it is easy to sample. A similar method is implemented in the LPF introduced in [2]. This topic is discussed in detail in Section 4.3.

4.2.1. Particle filter with uniform proposal distribution. When a non-informative (locally uniform) proposal distribution $q_T()$ is used in (6), the recursive expression for computation of the weights becomes

$$w_t^i \propto w_{t-1}^i p\left(\mathbf{y}_t, \mathbf{x}_t^i | \mathbf{x}_{t-1}^i\right), \tag{8}$$

DOI: 10.1002/acs

where

$$p\left(\mathbf{y}_t, \mathbf{x}_t^i | \mathbf{x}_{t-1}^i\right) \equiv f_w\left(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i\right) h_v\left(\mathbf{y}_t | \mathbf{x}_t^i\right).$$

Equation (8) puts in evidence that filtering is now performed by the joint likelihood[¶] of the observations and the current state conditioned on the prior state, instead of the measurement likelihood. This expression, which includes the prior PDF in the filtering stage of the filter, has been proposed in our previous work [36] under the designation of smooth kernel particle filter.

In terms of filter robustness, it is advantageous to use locally uniform proposals for a subset of the samples and to compute the corresponding importance weights according to (8). The main requirement for the success of this approach is the definition of an appropriate support of the uniform distribution. This topic is addressed in Section 4.4.3.

4.2.2. Particle filter with mixture proposal distribution. Consider a particle filter based on a mixture proposal distribution consisting of a set of samples drawn from the prior distribution (as used by the standard particle filter) and a fixed number of particles drawn from a uniform distribution with a support, S_u defined as a box in the state-space. The uniformly distributed samples — designated *extra particles* — are introduced to ensure sufficient sample diversity and avoid particle collapse. An expression for the computation of weights is derived formally in the sequel, which applies an importance ratio defined to take in account the incorporation of these extra particles. The resulting PF version, based on the mixture distribution, is designated *MPF*.

In principle, using an adaptive sample size, instead of a fixed number of particles, makes the filters computationally more efficient while maintaining its estimation performance. In practice, we did not find a significant difference between the two implementations in terms of execution speed. As such, our basic approach consists in the addition of a fixed number of samples drawn from a uniform distribution at the prediction stage of the filter.

The mixture distribution that results from using the two sets of weighted particles may be represented by[‖]

$$p_{T|T}(\mathbf{x}_t|\mathbf{y}_t) \approx \sum_{i=1}^{N_1} w_t^i \delta\left(\mathbf{x}_t - \mathbf{x}_t^i\right) + \sum_{i=N_1+1}^{N} w_t^{*i} \delta\left(\mathbf{x}_t - \mathbf{x}_t^i\right) s \tag{9}$$

where the weights w_t^i are computed as in the standard PF and $w_t^{*i} = \tilde{w}_t^{*i} \Big/ \sum_{j=N_1+1}^{N} \tilde{w}_t^{*j}$ represents the normalized weights of the extra particles; N is the total number of particles, N_1 is the number of samples drawn from the prior, and $N - N_1$ is the number of extra particles. In the present formulation, the update stage of the mixture of particles represented by (9) has two components — one given by expression (7), which applies to the standard particles and another defined by (8) that is used for the extra particles. The MPF algorithm is presented in the preceding text.

4.2.3. The prior-correction particle filter. According to the formal derivation of the MPF, expression (8) applies only to extra particles. However, experiments indicate that the application of this formula to all the particles used in the mixture results in highly improved robustness to outliers. This result is justified if we consider that multiplying by $f_w\left(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i\right)$ at the correction stage further attenuates the weights of outliers generated at the prediction stage of the filter. In terms of the IS method, the bias compensation mechanism implemented by the filter is modified to attenuate more efficiently the importance of particles that represent process model implausible states.

[¶]In rigor, the function $p\left(\mathbf{y}_t, \mathbf{x}_t^i | \mathbf{x}_{t-1}^i\right)$ can be interpreted as a likelihood in a parametric framework where $\mathbf{x}_{t-1}$ represents a fixed parameter but is a conditional PDF in the context of Bayesian inference where $\mathbf{x}_{t-1}$ denotes a random variable.

[‖]Formally, because the particle filter represents the PDFs using a sum of Dirac masses, the sample-based distribution and the target distribution can only be considered approximate in terms of their respective Lebesgue integrals. However, following common practice in the literature, we use the present expressions for simplicity of notation.

DOI: 10.1002/acs

As such, this computation of the weights contributes to mitigate the aforementioned problem of filter divergence caused by outliers with large measurement likelihoods associated to terrain symmetries. We designate this PF version as the *PPF*. The corresponding algorithm is

Algorithm4 : PPF

$\left[\left\{\mathrm{x}_k^i, w_k^i\right\}_{i=1}^N\right] = \mathrm{PPF}\left[\left\{\mathrm{x}_{k-1}^i, w_{k-1}^i\right\}_{i=1}^N, \mathrm{y}_k, N_s, N_e, K_{eff}, S_u\right]$

01. $N = N_s + N_e$
02. $N_T = N \cdot K_{eff}$
03. FOR $i = 1 : N_s$
04. $Draw \quad \mathrm{x}_k^i \sim p(\mathrm{x}_k^i|\mathrm{x}_{k-1}^i)$
05. END FOR
06. FOR $i = N_s + 1 : N$
07. $Draw \quad \mathrm{x}_k^i \sim Uniform(S_u)$
08. END FOR
09. FOR $i = 1 : N$
10. $w_k^i = w_{k-1}^i p(\mathrm{y}_k|\mathrm{x}_k^i) p(\mathrm{x}_k^i|\mathrm{x}_{k-1}^i)$
11. END FOR
12. S = SUM $\left[\left\{w_k^i\right\}_{i=1}^N\right]$
13. FOR $i = 1 : N$
14. $w_k^i = w_k^i \cdot S^{-1}$
15. END FOR
16. $\hat{N}_{\text{eff}} = 1/\text{SUM}\left[\left\{\left(w_k^i\right)^2\right\}_{i=1}^N\right]$
17. IF $\hat{N}_{\text{eff}} < N_T$
18. $\left\{\mathrm{x}_k^i, w_k^i, -\right\}_{i=1}^N = \text{RESAMPLE}\left[\left\{\mathrm{x}_k^j, w_k^j\right\}_{j=1}^N\right]$
19. END IF

4.2.4. Computational advantages and limitations of the mixture particle filter and prior-correction particle filter. In principle, the MPF and PPF filters are computationally more efficient because they implement a single weighted bootstrap operation at each iteration while the LPF and APF require two weighted bootstraps per iteration. However, this relative advantage of the MPF/PPF may only reveal significantly in high-dimensional spaces or when the number of particles used in some dimensions is very large. Because the MPF and PPF use mixture importance distributions that include uniforms, there are more samples being generated from a prior distribution without significant impact on the posterior. As a consequence, their implementation of the biasing mechanism used by importance sampling is less effective and may require a larger number of samples than the conventional implementations, which do not use non-informative priors. These relative advantages and limitations of the filters are assessed in the forthcoming benchmarking tests.

4.3. Likelihood particle filters

Likelihood PFs have been proposed in the literature under different designations (e.g., [2, 54, 60]). The unifying objective of this type of filters is to generate samples that correspond to a large likelihood of the measurements, thus avoiding the degeneracy of the particles. A direct implementation of this idea consists in sampling from the PDF that characterizes the observation model, $p(\mathbf{y}_t|\mathbf{x}_t^i) = h_v\left(\mathbf{y}_t|\mathbf{x}_t^i\right)$. When this expression is substituted for the importance sampling in (6), the resulting weight update expression is

$$w_t^i \propto w_{t-1}^i f_w\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i\right).$$

DOI: 10.1002/acs

Algorithm5 : **LPF**

$\left[\left\{\mathrm{x}_k^i, w_k^i\right\}_{i=1}^N\right] = \text{LPF}\left[\left\{\mathrm{x}_{k-1}^i, w_{k-1}^i\right\}_{i=1}^N, \mathrm{y}_k, N, N_T\right]$

FOR $i = 1 : N$
  REPEAT
    $Draw \quad \mathrm{s}_k^i \sim \hat{p}(\mathrm{s}_k|\mathrm{y}_k) \propto p(\mathrm{y}_k|\mathrm{s}_k)$
  UNTIL $\mathrm{s}_k^i \geqslant 0$
  IF Uniform $[0,1] > 1/2$
    $\mathrm{x}_k^i = \sqrt{\mathrm{s}_k^i}$
  ELSE
    $\mathrm{x}_k^i = -\sqrt{\mathrm{s}_k^i}$
  END IF
  $w_k^i = w_{k-1}^i p(\mathrm{x}_k^i|\mathrm{x}_{k-1}^i)\mathrm{x}_k^i$
END FOR
S = SUM $\left[\left\{w_k^i\right\}_{i=1}^N\right]$
FOR $i = 1 : N$
    $w_k^i = w_k^i \cdot S^{-1}$
END FOR
$\hat{N}_{\text{eff}} = 1/\text{SUM}\left[\left\{\left(w_k^i\right)^2\right\}_{i=1}^N\right]$
IF $\hat{N}_{\text{eff}} < N_T$
    $\left\{\mathrm{x}_k^i, w_k^i, -\right\}_{i=1}^N = \text{RESAMPLE}\left[\left\{\mathrm{x}_k^j, w_k^j\right\}_{i=1}^N\right]$
END IF

Algorithm6 : **GLPF**

$\left[\left\{\mathrm{x}_k^i, w_k^i\right\}_{i=1}^N\right] = \text{GLPF}\left[\left\{\mathrm{x}_{k-1}^i, -\right\}_{i=1}^N, \mathrm{y}_k, N, K_{eff}, S_u\right]$

$N_T = N \cdot K_{eff}$
FOR $i = 1 : N$
    $Draw \quad \mathrm{x}_k^i \sim Uniform(S_u)$
END FOR
FOR $i = 1 : N$
    $\nu_k^i = p(\mathrm{y}_k|\mathrm{x}_k^i)$
END FOR
S = SUM $\left[\left\{\nu_k^i\right\}_{i=1}^N\right]$
FOR $i = 1 : N$
    $\nu_k^i = \nu_k^i \cdot S^{-1}$
END FOR
$\left\{\mathrm{x}_k^i, -, -\right\}_{i=1}^N = \text{RESAMPLE}\left[\left\{\mathrm{x}_k^j, \nu_k^j\right\}_{j=1}^N\right]$
FOR $i = 1 : N$
    $w_k^i = p(\mathrm{x}_k^i|\mathrm{x}_{k-1}^i)$
END FOR
$\hat{N}_{\text{eff}} = 1/\text{SUM}\left[\left\{\left(w_k^i\right)^2\right\}_{i=1}^N\right]$
IF $\hat{N}_{\text{eff}} < N_T$
    $\left\{\mathrm{x}_k^i, w_k^i, -\right\}_{i=1}^N = \text{RESAMPLE}\left[\left\{\mathrm{x}_k^j, w_k^j\right\}_{j=1}^N\right]$
END IF

DOI: 10.1002/acs

This is the general expression used to update the importance weights of a LPF. We consider two different approaches to the implementation of this type of filters. First, we analyze the method proposed in the literature and present its generic form. Second, we propose a novel implementation that we designate as *generic likelihood particle filter*.

4.3.1. Current likelihood particle filters implementations. The LPF proposed in [2] uses an intermediate step where samples are generated from an auxiliary distribution $p(s_t|\mathbf{y}_t)$ through a rejection method, which discards particles $\mathbf{x}_t^i$ drawn from a uniform that correspond to null likelihoods (Algorithm 5). Using $s_t = h(\mathbf{x}_t)$, where $h(\mathbf{x}_t)$ is the deterministic component of the measurement model, ensures a large likelihood of the particles generated at the prediction stage. In this formulation, the utilization of the PDF $p\left(s_t^i|\mathbf{y}_t\right)$ instead of $p\left(x_t^i|\mathbf{y}_t\right)$ implies the application of the correction factor $p\left(x_t^i|\mathbf{y}_t\right)/p\left(s_t^i|\mathbf{y}_t\right)$ at the filtering stage.

The resulting weight update equation is thus

$$w_t^i \propto w_{t-1}^i f_w\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i\right)\frac{p\left(\mathbf{x}_t^i|\mathbf{y}_t\right)}{p\left(s_t^i|\mathbf{y}_t\right)}. \tag{10}$$

Notice that $p(s_t|\mathbf{y}_t)$ may become singular at some region of the state-space where application of the previous expression becomes unfeasible, but the proposed algorithm avoids this problem by application of the rejection method mentioned earlier.

It is straightforward to demonstrate that in the general case, the relation between the underlying distribution $p\left(x_t^i|\mathbf{y}_t\right)$ and the intermediate distribution $p\left(s_t^i|\mathbf{y}_t\right)$ is

$$p\left(s_t^i|\mathbf{y}_t\right) = \sum_{j=1}^{n}\frac{p\left(\mathbf{x}_{t,j}^i|\mathbf{y}_t\right)}{\left|h'\left(\mathbf{x}_{t,j}^i\right)\right|}, \quad (j = 1, ..., n)$$

where $\mathbf{x}_{t,j}^i, \quad (j = 1, ..., n)$ are the n real roots of function $h(\mathbf{x}_t)$ and $\left|h'\left(\mathbf{x}_{t,j}^i\right)\right| = \left|\frac{\partial h(\mathbf{x}_t)}{\partial \mathbf{x}_t}\right|$ is the determinant of the Jacobian of the general transformation $h(\mathbf{x}_t)$ (e.g., [62]). Hence, the weight updating expression (10) applies a correction factor that can be considerably complex and computationally demanding when applied to large particle sets.

Although the principle of the LPF is intuitive, it may yield poor results in the presence of terrain ambiguity; for example, a symmetric terrain will lead to a multi-modal posterior distribution that may cause filter divergence if the transition kernel used in the correction step is too smooth. Moreover, we observe that the method is not generally realizable, because the expressions for computation of the weights presented earlier may not be applicable if the measurement function $h(\mathbf{x})$ is not bijective.

It is worth noting at this point that the auxiliary particle filter mechanizes a method to draw samples that adjust well to the observed data and may be considered an indirect implementation of the likelihood sampling concept.

4.3.2. Derivation of the generic likelihood particle filter. Likelihood particle filters that sample from the likelihood at the prediction stage may not be practical to implement if the observation model cannot be represented by a well-behaved function. The likelihood PF version that we propose avoids this problem because it does not sample from the observation model or from a proportional to that density function and does not require the computation of a correction factor as in (10). The filter relies on the method of *importance resampling* (e.g., [50]) to draw a set of particles with large likelihoods and is generic because its applicability does not depend on the type of likelihood function used by the observation model. Additionally, the computational complexity of the filter does not depend on the complexity of the likelihood function. These characteristics justify its designation as the *GLPF*.

The GLPF draws samples from a proposal distribution proportional to $p(\mathbf{x}_t|\mathbf{y}_t)$, which is not conditioned on the previous state but only on the current measurement, $\mathbf{y}_t$. This sampling procedure is implemented considering the equality $p(\mathbf{x}_t|\mathbf{y}_t) = \frac{p(\mathbf{y}_t,\mathbf{x}_t)}{p(\mathbf{y}_t)} \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t)$ and taking in account

DOI: 10.1002/acs

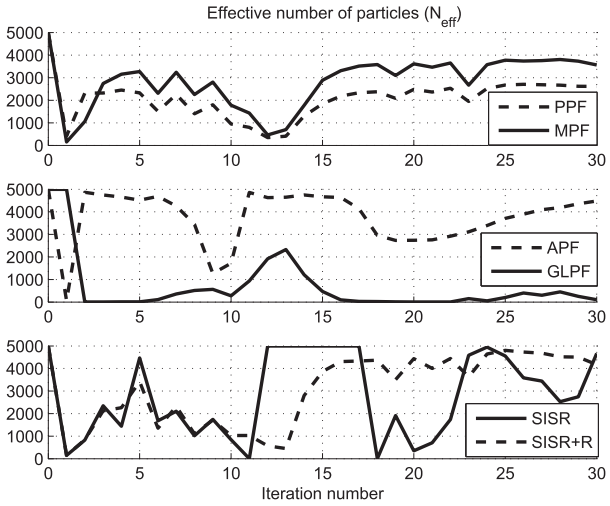

Figure 1. Comparison of the effective number of samples used by the different filters in the bearings-only problem ($N = 5000$).

that evaluation of the PDF up to a proportional constant is sufficient because the weights are normalized in the importance sampling formulation. The method, which is detailed in the GLPF algorithm presented previously, can be described as follows: draw a set of samples $\{\mathbf{x}_t^i\}_{i=1}^N$ from a locally uniform distribution as described in the next subsection; attribute to each sample a weight equal to the likelihood of the current measurement $\mathbf{y}_t$; finally, perform importance resampling on the sample set using the normalized set of weights $\{w_t^i\}_{i=1}^N$ as the acceptance probability.

Using the proposal density $q_T(.) = p(\mathbf{x}_t|\mathbf{y}_t) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t)$ with $p(\mathbf{y}_t|\mathbf{x}_t) \equiv h_v(\mathbf{y}_t|\mathbf{x}_t)$, (6) becomes:

$$w_t^i \propto w_{t-1}^i \frac{f_w\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i\right) p\left(\mathbf{y}_t|\mathbf{x}_t^i\right)}{p\left(\mathbf{y}_t|\mathbf{x}_t^i\right) p\left(\mathbf{x}_t^i\right)}.$$

If $\mathbf{x}_t^i$ is sampled from the non-informative distribution, then the previous expression simplifies to

$$w_t^i \propto w_{t-1}^i f_w\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i\right).$$

This likelihood PF implementation corresponds to the symmetric operation of the standard filter as could intuitively be expected in the context of importance sampling. Actually, the SISR draws samples from an approximation of the prior distribution and computes the weights using the measurement likelihood, while the GLPF samples from an approximation of the likelihood and weights the samples using the prior.

4.3.3. Computational advantages and limitations of the generic likelihood particle filter. The main advantage of the generic LPF is its ability to generate samples with very high measurement likelihoods while dispensing with the need to sample directly from a likelihood function. The GLPF avoids the complexity of the computations mentioned in Section 4.3.2 and, contrarily to the likelihood particle filters proposed in previous publications, enables the application to non-invertible likelihood functions. A negative effect of the method is the reduction of the effective number of samples as a consequence of resampling from the non-informative distribution; this problem is illustrated by the plot of the variable N_{eff} in Figure 1. Another disadvantage of the method is the application of importance resampling, which in principle is less efficient than rejection sampling as used in the likelihood particle filter proposed in [2]; see also [50]. Despite these problems, the benchmarked performance of the GLPF algorithm is similar to that obtained with the best performing algorithms.

DOI: 10.1002/acs

4.4. Utilization of non-informative prior distributions

4.4.1. Variance of particles generated by a uniform importance distribution. The undesirable effects of using a uniform importance distribution are discussed in [51]. Generic expressions for the variance of the weights and the effective number of samples used in the forthcoming analysis are also derived in [51] and discussed in [54]. To assess the impact of this type of non-informative distribution on filter efficiency, it is of interest to analyze the variance of the resulting particles that is employed to compute a measure of the effective sample size used by the filter: $N_{eff} = \frac{N}{1+N^2 \mathbf{v}ar_\pi(w_t^*)}$ where $\mathbf{v}ar_\pi(w_t^*)$ is the sample variance of the normalized weights[**] . In practical implementations, N_{eff} is approximated by $\hat{N}_{eff} = \left(\sum_{i=1}^{N} \left(w_t^i\right)^2 \right)^{-1}$ using the ensemble of weights available at a given iteration. Although other measures of the effective number of samples have been proposed in the literature (e.g., [32, 57] and additional references therein), this formula for $\hat{N}_{eff}$ is used in the vast majority of PF implementations.

Defining the proposal density as a uniform PDF, $\pi_\theta = \lambda$, and using the formula of weight update (6) and the equality $p_\theta\left(\mathrm{x}_t, \mathrm{y}_t|\mathrm{x}_{t-1}^i\right) = p_\theta\left(\mathrm{y}_t|\mathrm{x}_t^i\right) p_\theta\left(\mathrm{x}_t^i|\mathrm{x}_{t-1}^i\right)$, the expression for the computation of the variance of the weights is given by

$$
\begin{aligned}
Var_{\pi_\theta}\left(w_t^i\right) &= \int_S \left[\left(w_{t-1}^i\right) \frac{p_\theta\left(\mathrm{y}_t|\mathrm{x}_t^i\right) p_\theta\left(\mathrm{x}_t^i|\mathrm{x}_{t-1}^i\right)}{\pi_\theta\left(\mathrm{x}_t|\mathrm{x}_{0:t-1}^i, \mathrm{y}_{1:t}\right)} \right]^2 \pi_\theta\left(\mathrm{x}_t|\mathrm{x}_{0:t-1}^i, \mathrm{y}_{1:t}\right) d\mathrm{x}_t \\
&\quad - \left[\int_S \left(w_{t-1}^i\right) \frac{p_\theta\left(\mathrm{y}_t|\mathrm{x}_t^i\right) p_\theta\left(\mathrm{x}_t^i|\mathrm{x}_{t-1}^i\right)}{\pi_\theta\left(\mathrm{x}_t|\mathrm{x}_{0:t-1}^i, \mathrm{y}_{1:t}\right)} \pi_\theta\left(\mathrm{x}_t|\mathrm{x}_{0:t-1}^i, \mathrm{y}_{1:t}\right) d\mathrm{x}_t \right]^2 \\
&= \left(w_{t-1}^i\right)^2 \int_S \frac{\left[p_\theta\left(\mathrm{y}_t|\mathrm{x}_t^i\right) p_\theta\left(\mathrm{x}_t^i|\mathrm{x}_{t-1}^i\right)\right]^2}{\pi_\theta\left(\mathrm{x}_t|\mathrm{x}_{0:t-1}^i, \mathrm{y}_{1:t}\right)} d\mathrm{x}_t - \left(w_{t-1}^i\right)^2 \int_S p_\theta\left(\mathrm{y}_t|\mathrm{x}_t^i\right) p_\theta\left(\mathrm{x}_t^i|\mathrm{x}_{t-1}^i\right) d\mathrm{x}_t \\
&= \left(w_{t-1}^i\right)^2 \left[\int_S \frac{\left[p_\theta\left(\mathrm{x}_t|\mathrm{y}_t, \mathrm{x}_{t-1}^i\right) p_\theta(\mathrm{y}_t|\mathrm{x}_{t-1}^i)\right]^2}{\lambda} d\mathrm{x}_t - p_\theta(\mathrm{y}_t|\mathrm{x}_{t-1}^i)^2 \right] \\
&= A\left[w_{t-1}^i p_\theta(\mathrm{y}_t|\mathrm{x}_{t-1}^i)\right]^2 \left[\int_S \left[p_\theta\left(\mathrm{x}_t|\mathrm{y}_t, \mathrm{x}_{t-1}^i\right)\right]^2 d\mathrm{x}_t - 1 \right],
\end{aligned}
$$

where S denotes the entire state space and $A = \frac{1}{\lambda}$ is the size of the support of the uniform distribution. The former expression shows that the variance of the weights increases linearly with the size of the support of the uniform proposal density. This fact should be considered to define the area where the particles are uniformly distributed, when using uniform priors or mixture distributions that includes uniforms. The topic is addressed in the remainder of this section.

4.4.2. Non-informative prior distributions – main problems and proposed solutions. The utilization of non-informative priors for Bayesian inference is discussed at length in the literature. See, for example, [63] and [9], for a theoretical introduction to the subject and related practical problems. In simple terms, a prior density function of a given parameter can be considered non-informative if it is approximately constant over the region in which the likelihood function is significant. In such case, the influence of the prior in the computation of the posterior is minimal. It is common practice to use uniform distributions to represent the lack of knowledge with respect to some parameter of interest. The approach is subjected to criticism among other reasons because the uniform distribution is not invariant under nonlinear transformations of the parameters, that is, the uniform prior

[**] As noted in [54], the literature often defines $N_{eff} = \frac{N}{1+\mathbf{v}ar_\pi(w_t^*)}$, which is incorrect. We recall also that the variance used in the theoretical definition of N_{eff} is computed at some instant, t for a given Monte Carlo chain represented by the superscript 'i' in w_t^i, instead of the variance of the set of weights in the different chains, $\left\{w_t^i\right\}_{i=1}^N$.

DOI: 10.1002/acs

may be non-informative under a given parametrization but not necessarily non-informative when a change of variables involving a nonlinear transformation is applied. In principle, the representation of ignorance should be invariant under injective transformations of the parameter of interest. Among the alternative, non-informative distributions proposed in the literature, the prior proposed by Jeffreys based on the Fisher information matrix has the property of being locally uniform and invariant to nonlinear transformations (e.g., [9] and [64]). Besides, it provides a non-subjective, automated method for the definition of a non-informative prior for any parametric model.

In the present work, the bi-dimensional parameters of interest correspond to the two-dimensional localization of a vehicle and the uniform priors on these parameters are applied with two objectives: (1) in the MPF and PPF filters, extra particles, independent of prior knowledge, are sampled from a uniform distribution on the localization space at the prediction stage of the filter with the objective of maintaining particles diversity; (2) in the LPF anf GLPF filters, samples drawn from a uniform distribution are used at an intermediate step to implement an importance sampling density that is proportional to the likelihood of the measurements and independent from prior knowledge.

Our main concern in this context is the definition of a suitable support for the uniform distributions used by the filter. Actually, unless properly parameterized, the utilization of uniform distributions may have undesirable effects. Sampling from a uniform distribution on the state space, especially in the case of high-dimensional spaces or large-scale environments as used in terrain navigation, may require an excessively large number of particles making the filter excessively demanding computationally. Without an appropriate criterion to define a compact support of the distribution, uniform samples may include an excessively large number of implausible particles according to the process model. Although these particles can be filtered adequately as proposed in the filters introduced here, the rejection of a large number of samples results in loss of particles diversity. Moreover, as shown before, the variance of the weights increases proportionally to the size of the support causing the reduction of the effective sample size. Thus, using a large support of the uniform distribution may reduce severely the computational efficiency of the filter. Conversely, if the support utilized is too small or incorrectly centered, the prior may not ensure sufficient spreading of extra particles or it may not be locally uniform relatively to the likelihood thus violating the main assumption of the filter formulation (in the case of likelihood particle filters).

Given the absence of references in the navigation literature regarding this subject, we propose an automated criterion to define the subset of the state-space that can be used for sampling from uniform distributions in this type of filters. The procedure is derived from the Jeffreys' method mentioned earlier and addresses the two issues discussed formerly: defining supports of uniform distributions computationally amenable to practical implementations and ensuring local uniformity of the distribution. It uses a prior $\pi(\theta)$ on the parameters θ that is based on the Fisher information matrix according to the following general formulation:

$$\pi(\theta) \propto |J(\theta)|^{\frac{1}{2}}$$

where $J(\theta)$ is the Fisher information matrix relative to the parameter θ, whose elements, as derived in [65], are

$$J_{ij}(\theta) = -E\left[\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log p(\mathrm{y}|\theta)\right].$$

In the case of TAN, the non-informative prior represents the distribution of the two-dimensional position of the vehicle, $\mathrm{x_t}$. Similary to the formulation used in [36], the Fisher information is given by

$$J(\mathrm{x}_t) = -E\left\{\nabla_{\mathrm{x}_t}\left[\nabla_{\mathrm{x}_t} \log p(\mathrm{y}_t|\mathrm{x}_t)\right]\right\} = E\left\{H_t^\top R_t^{-1} H_t\right\}, \tag{11}$$

where $H_t^\top = \left[\nabla_{\mathrm{x_t}} h_t^\top(\mathrm{x_t})\right]$ represents the Jacobian of the measurement function $h_t(.)$ evaluated at state x_t and R_t is the time-varying measurement noise intensity matrix. Notice that in the non-parametric case, that is, when the parameter corresponding to the 2D position is not deterministic, the mathematical expectation used previously is computed with respect to the distribution of the random variable x_t. This computation can be done by Monte Carlo integration as suggested in [24].

DOI: 10.1002/acs

4.4.3. Implementation of the adaptive support of a uniform prior. To define the supports of the uniform distributions used by the filters, we apply the measure of terrain information introduced by [24] based on the Fisher information matrix. Instead of using the determinant of the Fisher information matrix as proposed in the Jeffreys' method, we use its trace as a scalar measure of the local information extracted from the terrain. The derivation of this measure is summarized later; please refer also to [36].

Let S_u be an arbitrary region with area A inside the map. Assume a non-informative prior $p(\mathbf{x})$ on the initial position of the vehicle approximated by the uniform distribution

$$p(\mathbf{x}) = \begin{cases} A^{-1} & \mathbf{x} \in S_u \\ 0 & \text{elsewhere} \end{cases} \tag{12}$$

and a measurement obtained according to

$$\mathbf{y}_t = h(\mathbf{x}_t) + \eta(\mathbf{p_t}), \tag{13}$$

where the additive scalar measurement noise η is Gaussian with zero mean and variance $\sigma^2 = \lambda$. The joint density of the states and the measurement is

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) = \begin{cases} A^{-1}\mathcal{N}(\mathbf{y}; h(\mathbf{x}), \lambda) & \mathbf{x} \in S_u \\ 0 & \text{elsewhere} \end{cases} \tag{14}$$

The Fisher information matrix is

$$J = \frac{1}{\lambda} E\left(\nabla_{\mathbf{x}} h(\mathbf{x}) \nabla_{\mathbf{x}}^{\top} h(\mathbf{x})\right) = \frac{1}{\lambda A} \int_{S_u} \nabla_{\mathbf{x}} h(\mathbf{x}) \nabla_{\mathbf{x}}^{\top} h(\mathbf{x}) dx. \tag{15}$$

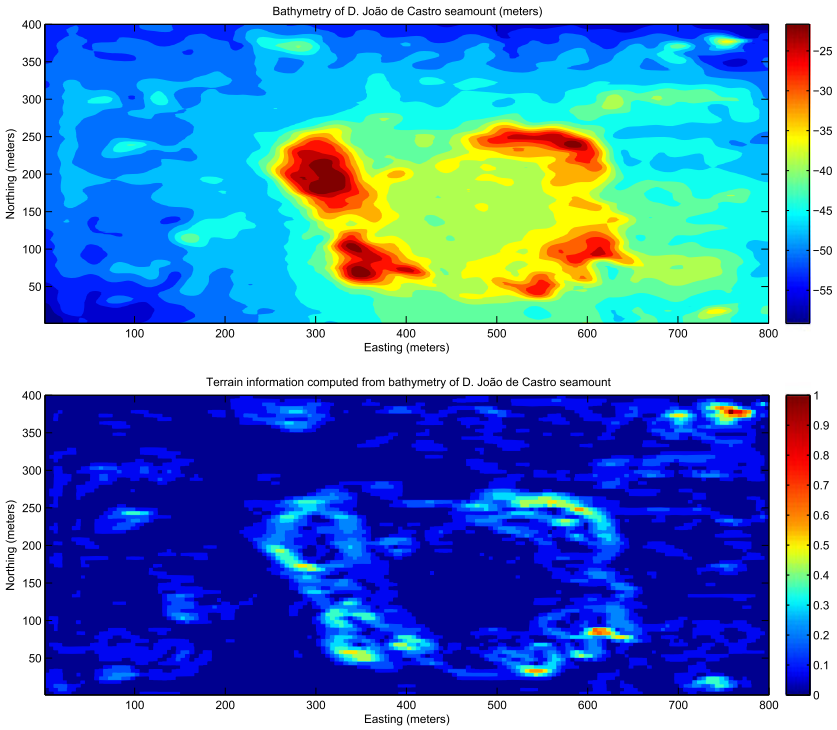

Figure 2. Topographic map (top) and normalized terrain information (bottom) computed from the bathymetry of the D. João de Castro seamount, Azores, PT. [Colour figure can be viewed at wileyonlinelibrary.com]

DOI: 10.1002/acs

Using a set of points $\mathbf{x}_i$ uniformly distributed over the area $\mathcal{S}$, it is possible to obtain a scalar measure of the terrain information, using a Monte Carlo approximation to the integral in (15) as follows:

$$I = \sqrt{tr\, J} \simeq \frac{1}{\sigma}\sqrt{\frac{1}{N}\sum_{i}^{N}\left\|\nabla_{\mathbf{x}_i} h(\mathbf{x}_i)\right\|^2}. \tag{16}$$

These concepts are illustrated in the sequel, using the bathymetric map represented in Figure 2 (top); the corresponding terrain information map is shown in the same figure (bottom).

Given a local measure of the terrain information, I, we define a locally uniform distribution centered on the estimated 2D position $\hat{\mathbf{x}}_t = [\hat{x}_t, \hat{y}_t]$, with the rectangular support

$$S_u = (x_t, y_t) \in \mathbb{R}^2 : \left(\hat{x}_t - \frac{a}{2}\right) \leqslant x_t \leqslant \left(\hat{x}_t + \frac{a}{2}\right), \left(\hat{y}_t - \frac{b}{2}\right) \leqslant y_t \leqslant \left(\hat{y}_t + \frac{b}{2}\right)$$

and with probability density $\pi(\mathbf{x}_t) = I^* \propto I$, where I^* is the normalized terrain information, $I^* \in [0, 1]$. By definition of the probability density of a uniform distribution, the pair of constants (a, b) that parametrize the support must verify

$$ab = \frac{1}{I^*}.$$

Assuming that the terrain information is locally isotropic, the uniform distribution can be defined over squares with area a^2 such that

$$a = \frac{1}{\sqrt{I^*}}, \quad I^* \neq 0.$$

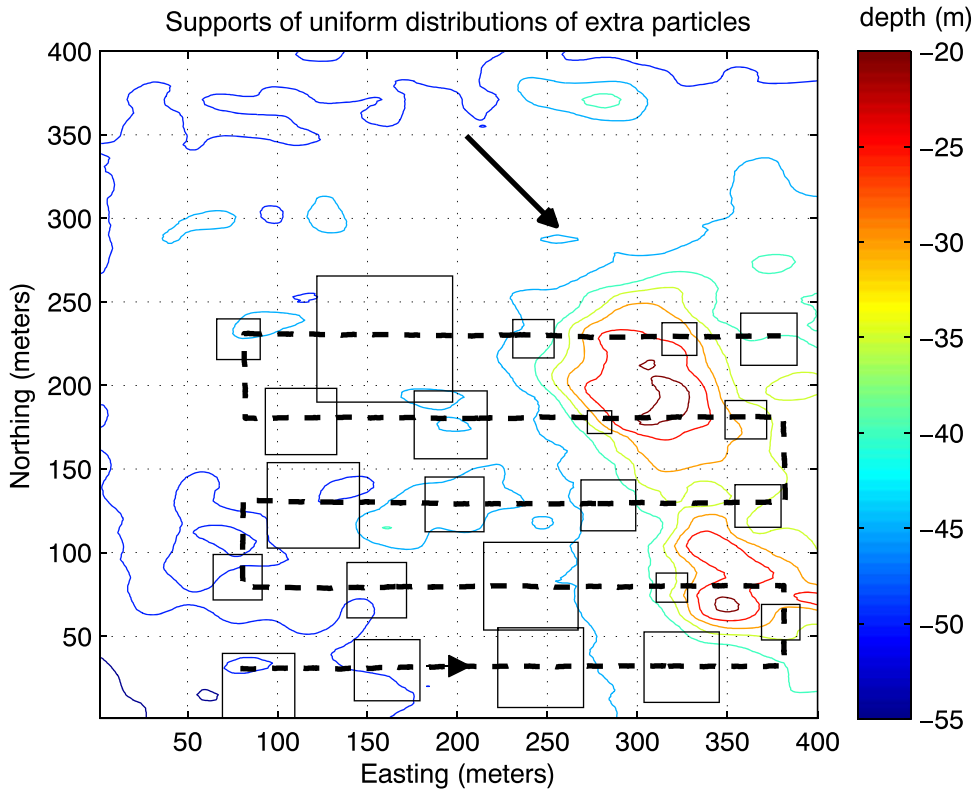

Figure 3. Localization of the trajectory used in simulations with the bathymetric map of D. João de Castro seamount. The large arrow indicates the direction of the ocean current whose velocity vector is estimated by the filter. The rectangles represent the areas of the supports of the uniform distributions of the extra particles used by the prior-correction particle filter, computed along the trajectory by an adaptation of Jeffreys' method (shown here for a subset of regularly spaced filter iterations). [Colour figure can be viewed at wileyonlinelibrary.com]

DOI: 10.1002/acs

In practice, we use

$$a = \begin{cases} \min\left[\sqrt{\frac{dA}{I^*}}, a_{\max}\right] & I^* > 0 \\ a_{\max} & I^* = 0 \end{cases}$$

where dA is the area of the elementary patch of the terrain used in the computation of I and $a_{\max}$ is the maximum area of the support of the distribution that must be enclosed by the prior map.

Figure 3 illustrates the application of this method to the solution of the TAN problem using the PPF filter with real bathymetric data of the *D.João de Castro* seamount. Notice that in this example the number of particles is constant along the simulation but in general the efficacy of the approach may require an adaptive number of uniform samples drawn at each iteration of the filter. The squares plotted over the contour map of the terrain show the excellent performance of the method which adapts the sizes of the areas to the excitation of the terrain, as desired.

5. PARTICLE FILTER BENCHMARKING

5.1. Introductory notes on benchmarking methods and metrics

To assess the performance of the novel particle filter formulations in different contexts, we applied them to solve two estimation problems that have been used frequently in the literature for filter benchmarking – a one-dimensional, nonlinear estimation example and the bearings-only tracking problem; both problems are addressed in [2, 3, 52, 66] and [45]; the tracking problem is also studied in [67]. In the next subsection, we present the results of Monte Carlo simulations where we apply the formerly discussed methods to solve the above nonlinear estimation problems. The different metrics used for performance evaluation in terms of the estimation error are defined in Section 1.2. In order to assess the computational effort imposed by the different algorithms, we employ as metrics the maximum amount of random-access memory (RAM) used during the simulations and the average time of CPU spent in a filter iteration (computed in an ensemble of 100 Monte Carlo runs). The tests were executed with a Toshiba Satellite A100 laptop with the following characteristics: CPU Intel T2400@1.83GHz; 2GB of RAM; operating system Windows XP Pro SP3.

5.1.1. Asymptotic filter performance. The use of sequential Monte Carlo methods or particle filters is justified theoretically by the law of large numbers, that is, the computed estimates are shown to converge almost surely to the quantity of interest as the number of particles tends toward infinity; see, for example, the theoretical demonstrations in [48]. In practical implementations, a limited number of particles, depending among other factors on the dimension of the state space, is sufficient to guarantee the convergence of the estimators. For a given application, it is important to assess the asymptotic performance of the filter, that is, the convergence of the computed estimates with the increasing number of particles. We consider this analysis more comprehensive than the presentation of a single measure of the estimation accuracy for a fixed number of samples. In this section, we provide simulations results that illustrate the asymptotic performance of SISR, APF, LPF, MPF, PPF, and GLPF algorithms.

5.1.2. Effective number of particles. The number of samples (particles) used by a particle filter is a fundamental parameter of the filter set-up that justifies the asymptotic filter performance analysis mentioned above. A related, time-varying measure that is frequently related to the performance of a particle filter is the *effective number of particles*, $\hat{N}_{eff}$, which has been defined in paragraph 4.4.1. This number measures approximately the proportion of particles that have significant weights at a given iteration of a particular filter set-up; it depends on the estimation algorithm and the estimation context. In the present study, we show that, despite its importance for the performance of the filter,

[††]It must be recognized that centering the support S_u uses the prior information conveyed by $\hat{\mathbf{x}}$ although the nominal prior distribution is non-informative.

DOI: 10.1002/acs

this number in isolation cannot be used to assess filter efficiency and should not be employed as a benchmarking parameter in the comparison of different filters. As shown in paragraph 5.3.2, for some classes of filters, the estimation context may lead to large values of $\hat{N}_{eff}$ that are not associated with good filter performance. Conversely, a filter may achieve superior estimation results despite using a considerably smaller effective number of particles. A clear illustration of the inadequacy of $\hat{N}_{eff}$ to indicate the convergence properties of a particle filter can be found in [68] that addresses specifically the topic of online filter consistency check.

5.2. One-dimensional nonlinear problem

5.2.1. Problem formulation and simulations' setup. The one-dimensional problem presented succeedingly is based on a univariate non stationary growth model used in econometrics ([69]) that has been employed frequently in the literature for filter benchmarking (e.g., [2, 52]). The state-space equations are as follows:

$$\begin{aligned} \mathbf{x}_t &= 0.5\mathbf{x}_{t-1} + 25\mathbf{x}_{t-1}/\left(1+\mathbf{x}_{t-1}^2\right) + 8\cos\left[1.2(t-1)\right] + w_t \\ \mathbf{y}_t &= \mathbf{x}_t^2/20 + v_t \end{aligned} \tag{17}$$

where the process and measurement models are defined by $w_t \sim N(0,\sigma_w)$ and $v_t \sim N(0,\sigma_v)$ with $\sigma_w = \sqrt{10}$ and $\sigma_v = 1$, respectively.

The system is highly nonlinear both in the process and the measurement model and includes a term, independent of the state that can be interpreted as time-varying noise. Note also that for $\mathbf{y}_t < 0$, the measurement likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ is unimodal and symmetric about zero but for $\mathbf{y}_t > 0$, it is bimodal and symmetric about zero. These characteristics make the problem more difficult to solve using conventional methods such as the extended Kalman filter.

Results representative of the performance of different estimation algorithms applied to this problem have been reported among others in the following publications: the bootstrap filter in [52]; the extended Kalman filter, the Gaussian particle filter, the unscented Kalman filter, and the SISR algorithm, in [66]; the extended Kalman filter, the SIR, the auxiliary particle filter, and the likelihood particle filter in [2]; the SIR and the SISR results are compared in [45].

We selected for benchmarking the filters that provided the best performance according to the results reported in the aforementioned papers. A series of benchmarks obtained in our tests with these algorithms is compared below with the corresponding results of the novel filter versions.

5.2.2. Simulations results and discussion. In Figure 4, we plot the approximate PDFs used by the SISR and the PPF for one iteration of the filter characterized by a multimodal measurement likelihood. The prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the posterior $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_t)$ PDFs shown are obtained from the corresponding sample sets using a kernel density estimator based on a Gaussian kernel with an adaptive bandwidth (e.g., [70]). The likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ is evaluated for the support of the particles,

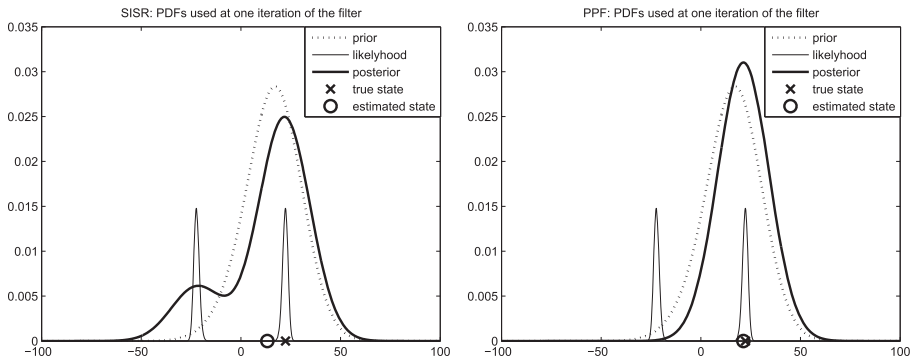

Figure 4. Kernel density estimates of the probability density functions (PDFs) used in one iteration of the sequential importance sampling + resampling (SISR) (left) and prior-correction particle filter (PPF) (right) filters applied to the one-dimensional problem.

DOI: 10.1002/acs

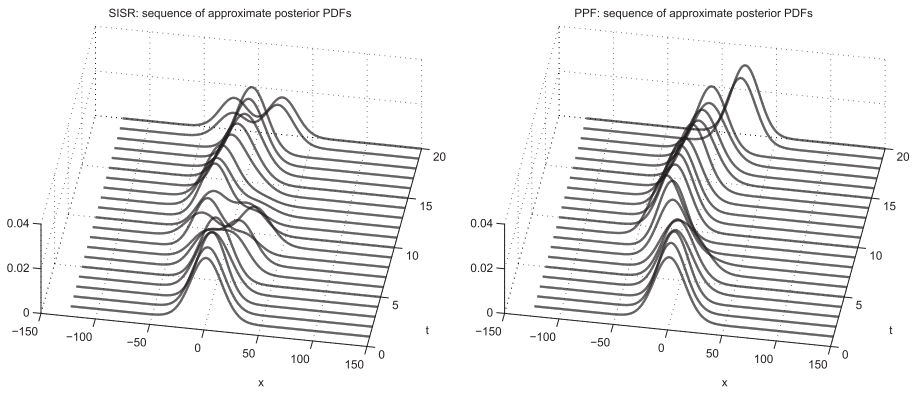

Figure 5. Sequence of approximate probability density function (PDFs) used in iterations 1–20 by the sequential importance sampling + resampling (SISR) (left) and prior-correction particle filter (PPF) (right) filters applied to the one-dimensional problem.

using its analytic expression. Figure 4 illustrates a fundamental difference in the operation of the PPF and the standard PF. As mentioned before, the PPF updates each weight at the correction stage multiplying its value at the previous iteration by the corresponding transition probability and by the current measurement likelihood. The standard PF updates the weights only by the measurement likelihood. Hence, at the filtering stage, this filter attenuates only the weights of particles corresponding to unlikely observations while the PPF attenuates the weights of process model outliers as well as observation-implausible states. The posterior distributions used by the SISR are clearly more diffuse and skewed than the corresponding distributions used by the PPF; see the sequence of PDFs used by these filters in Figure 5.

As noted in [45], the standard PF manifests a lack of robustness that results from a mismatch between the prior predictive distribution (used as the proposal) and the posterior distribution of the states conditioned on the new measurement. A partial mismatch of this type can be observed in the approximate PDFs depicted in Figure 4; it corresponds to a violation of a fundamental assumption of importance sampling because the support of the prior does not include the support of the posterior. This deficiency occurs in many other iterations of the SISR in the same simulation.

In Figure 6, we plot the RMSE for 100 MC runs ($RMSE_t^{100}$) obtained with all the filters. These plots show the superior performance of the PPF algorithm in terms of estimation error.

The asymptotic performance of the filters in terms of the $RMSE^{100}$ is represented in Figure 7. We notice that the LPF and the APF only perform better than the other versions when the number of particles is very small (equal or less than 50). For larger sample sizes, the standard PF performs better than those filters and is comparable to the MPF algorithm that we introduce. The novel PPF algorithm presents the best performance overall in terms of estimation error. The results presented here show that the smallest estimation error is achieved by all the filters analyzed when the number of particles is equal or larger than 1000. As such, we use this number of particles to benchmark the performance of the different filters in terms of estimation accuracy.

5.3. Bearings-only tracking problem

5.3.1. Problem formulation and simulations setup. Bearings-only tracking is a well studied problem in defense and other engineering applications. It is addressed using a common formulation in the references presented at the beginning of this section. The topic is also studied in [71] and [72], although with a different set-up. The problem formulation used here is similar to the one presented in [66][‡‡] . A target moves in the $x - y$ plane according to the process model

$$\mathbf{x}_t = \Phi \mathbf{x}_{t-1} + \Gamma \mathbf{w}_t \tag{18}$$

[‡‡]In order to make the presentation of the results more intuitive and avoid logarithmic scales in some plots, we use a modified scale by multiplying the distances and velocities by a factor of 10^3 and the corresponding covariances by 10^6. Following the practice adopted in the referenced literature, the units used for 2D localization are arbitrary.

DOI: 10.1002/acs

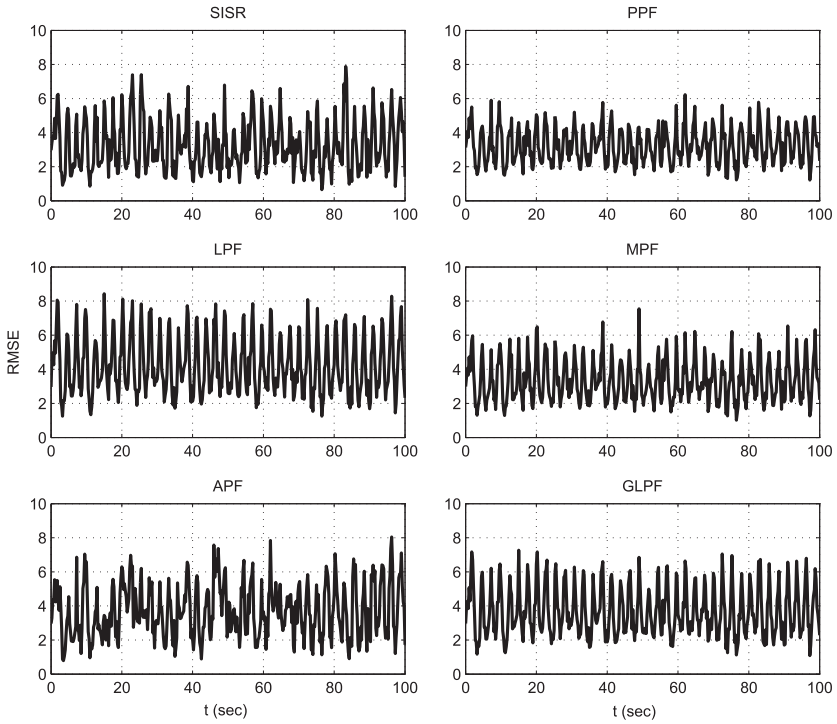

Figure 6. Estimation error ($RMSE_t^{100}$) obtained with the different algorithms applied to the one-dimensional problem.

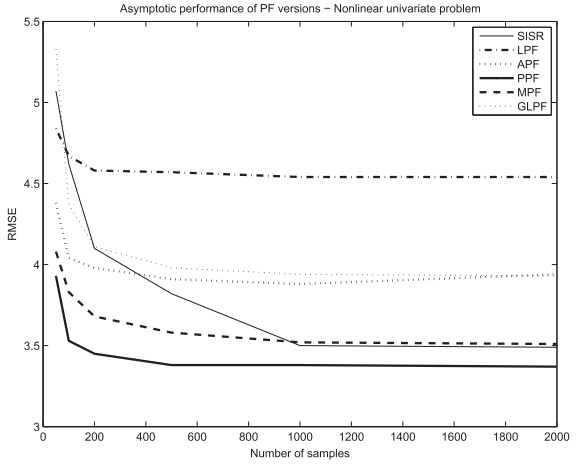

Figure 7. Comparison of the asymptotic performance of different particle filter (PF) versions in terms of the root-mean-square error ($RMSE^{100}$), applied to the one-dimensional estimation problem.

where $\mathbf{x}_t = [x_t, y_t, v_x, v_y]_t^\top$, $\mathbf{w}_t = [w_x, w_y]_t^\top$,

$$\Phi = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \textit{and} \quad \Gamma = \begin{bmatrix} 0.5 & 0 \\ 0.5 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

Here, Δt denotes the filter iteration time. The process noise is Gaussian with zero mean and uncorrelated in $\mathbf{x}$ and $\mathbf{y}$, that is, $\mathbf{w} \sim N(0, \sigma_w^2 I_2)$, where I_2 represents the identity matrix. The initial state estimate is distributed according to $\mathbf{x_0} \sim N(\mu_0, P_0)$.

DOI: 10.1002/acs

A static observer positioned at the origin of the coordinates system takes noisy measurements of the target bearing, according to the measurement model

$$\mathbf{y}_t = \tan^{-1}(y_t/x_t) + \mathbf{v}_t \tag{19}$$

where the measurement noise is a Gaussian, zero-mean sequence, that is, $\mathbf{v}_t \sim N(0, \sigma_v^2)$.

The simulations are parameterized as follows. The prior parameters are

$$\mu_0 = [-0.05, 0.7, 0.001, -0.055]^\top$$

and

$$P_0 = diag(0.1, 0.1, 0.005, 0.01).$$

The iteration time of the filters is $\Delta t = 1$. The process and measurement noise standard deviations are $\sigma_w = 0.001$ and $\sigma_v = 0.005$, respectively.

For benchmarking purposes, we compare the results obtained by the new particle filter versions with those achieved by the best performing algorithms according to the aforementioned literature – the SISR and the APF. Because the standard SISR algorithm performs poorly in comparison with the new filter versions, we implemented the SISR with the *roughening* procedure introduced in [52]; the results obtained with this version are presented under the designation sequential importance sampling + resampling (SISR+R). With the objective of making the tests more exhaustive, we extend the simulation time from 24 iterations (used in the reference literature) to 30.

5.3.2. Simulations results and discussion. A plot of the true target trajectory and the estimated positions, as well as the 2σ bounds estimated by the filter is presented in Figure 8a.

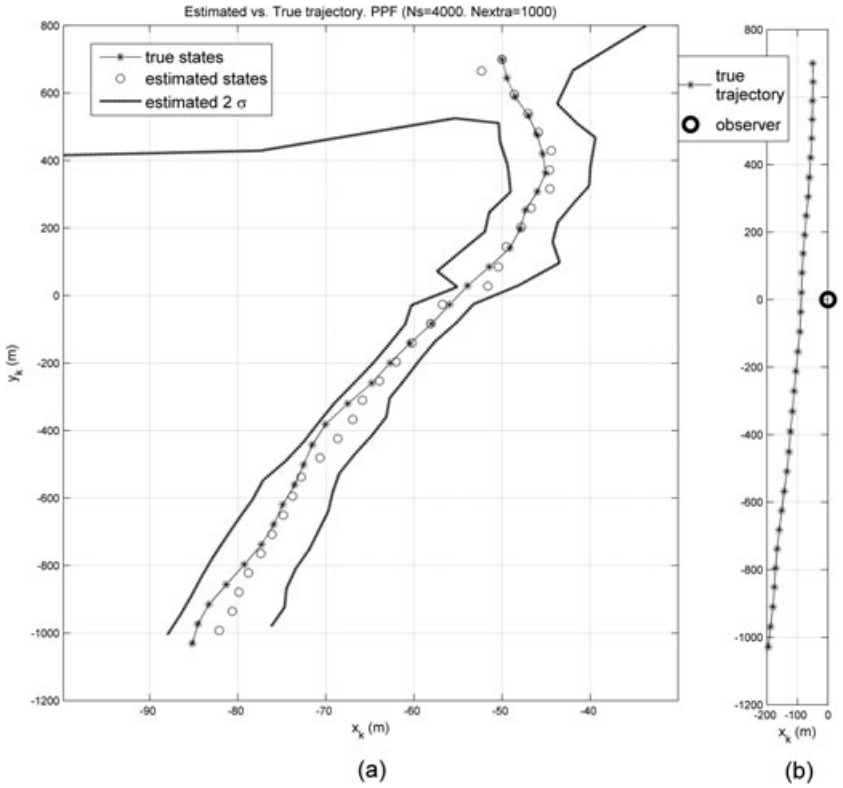

Figure 8. Bearings-only problem: (a) real and estimated target trajectory (not to scale) and 2σ bounds estimated by the prior-correction particle filter (PPF) filter; (b) true-scale location of observer and target.

DOI: 10.1002/acs

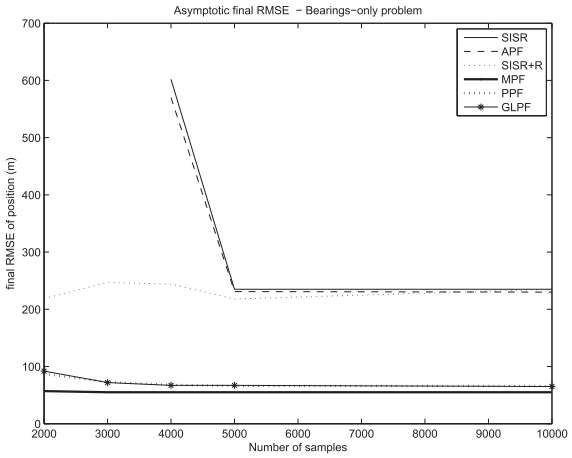

Figure 9. Comparison of the asymptotic performance of different particle filters (PF) versions in terms of the final estimation error ($RMSE_{30}^{100}$) of position in the bearings-only problem. Note that the sequential importance sampling + resampling (SISR) and auxiliary particle filter (APF) algorithms diverge with $N < 4000$.

Table I. RMSEs of position estimation by different filter versions as a function of the number of particles. Bearings-only problem (results from 100 Monte Carlo runs).

Filter version	RMSE	$N = 2000$	$N = 3000$	$N = 4000$	$N = 5000$	$N = 10,000$
SISR	Final	− − −	− − −	602	234	235
	$t = 24$	− − −	− − −	455	184	185
	% div.	100	81	0	0	0
SISR+R	Final	219	247	244	218	235
	$t = 24$	162	177	176	158	185
	% div.	0	0	0	0	0
APF	Final	− − −	− − −	570	231	230
	$t = 24$	− − −	− − −	415	197	190
	% div.	100	38	0	1	0
MPF	Final	57	55	55	54	54
	$t = 24$	39	37	37	36	37
	% div.	0	0	0	0	0
PPF	Final	87	73	68	66	66
	$t = 24$	60	50	46	44	23
	% div.	0	0	0	0	0
GLPF	Final	92	72	65	67	61
	$t = 24$	94	77	75	75	72
	% div.	0	0	0	0	0

APF, auxiliary particle filter; GLPF, generic likelihood particle filter; MPF, mixture particle filter; PPF, prior-correction particle filter; SISR, sequential importance sampling; SISR+R, sequential importance sampling + resampling; RMSE, root mean squared error.

The asymptotic performance of each filter is summarized by the final RMSE of position, that is, $RMSE_{30}^{100}$, which is shown in Figure 9. To facilitate the comparison with the results published in previous papers that limit the duration of simulations to $t = 24$, we also include in Table I the RMSE obtained for that time index. No results are presented of simulations where the filter diverges in more than 25% of the MC runs. Based on the results illustrated earlier, the setup with $N = 5000$, which ensures the best positioning accuracy with all the algorithms, is adopted to illustrate the filters performance in the following figures.

To enable a more detailed comparison of individual filter performance, plots of the RMSE obtained by the filters for a fixed number of particles are shown in Figure 10. The approximate distributions of position estimation errors and the corresponding $RMSE_t^{100}$ obtained with the dif-

DOI: 10.1002/acs

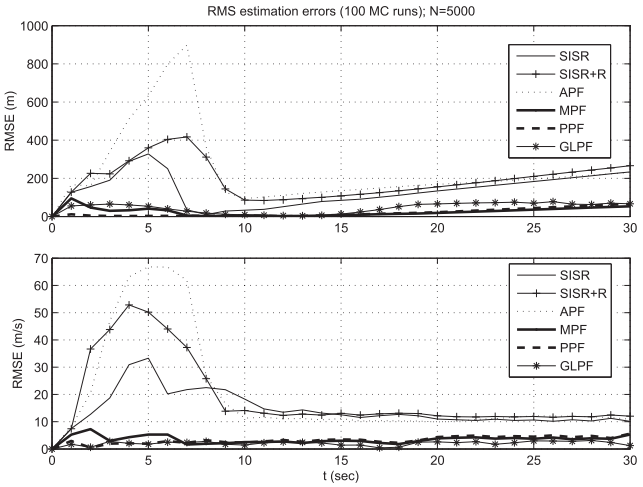

Figure 10. Comparison of the of position and velocity estimation errors ($RMSE_t^{100}$) obtained with different PF versions applied to the bearings-only problem ($N = 5000$).

ferent filters are shown in Figures 11 and 12. Figure 1 compares the evolution of $\hat{N}_{eff}$ along the simulation for the six algorithms. The plots show that the largest values of $\hat{N}_{eff}$ do not correspond necessarily to the filters that present the best estimation performance. Detailed illustrations of the behavior of the SISR and PPF filters along the simulation, represented by the true and estimated states, and the predicted and filtered particles, are available online for download at this URL.

The RAM required to execute the algorithms is always below 1 MB even with the maximum number of particles (10,000). As shown in Figure 13, the CPU time per iteration increases linearly with the increasing number of particles in all the filters. These metrics are summarized in Table II for the different filters using 5000 particles.

5.4. Benchmarking conclusions

The results obtained in the previous sections put in evidence the superior performance of the novel particle filter algorithms that we propose, in terms of estimation accuracy and robustness to outliers, relatively to the standard PF and other well-known versions. This improved performance of the MPF and PPF algorithms can be attributed in part to the type of proposal distribution used by these filters. Actually, the mixture distribution employed by these filters uses a support that can be much larger than that of the prior used in other versions. This contributes to maintaining the diversity of the particles and to reduce the effect of particle degeneracy. However, the main factor contributing to the good performance of both filters and to that of the PPF in particular is attributed to the incorporation of the prior conditional probability in the updating stage of the filters. Using this factor to update the full set of weights contributes to the rejection of outliers even in the presence of ambiguity in the measurement model. This characteristic of the filter is key to its successful application to TAN.

As shown in Figure 13, all the filters execute one iteration in less than 60 ms, a result that proves to be compatible with most real-time tracking applications such as the bearings-only problem. The SISR is faster than all the other filters; this result was expected considering the relative simplicity of the algorithm. However, in the range of sample sizes under analysis (from 2000 to 10,000 particles) the execution times of the SISR, SISR+R, MPF, PPF, and GLPF filters do not differ significantly; the maximum difference in terms of time per iteration among these filters is 5 ms for a sample size of 10,000 particles. For 5000 particles, which is the sample size considered in the RMSE-based analysis of filter performance, the differences in execution times are negligible even for time-critical applications. The APF presents the largest execution times per iteration. This result is a consequence of the structure of the algorithm, which is executed in two stages (the first one executed in instructions 1–5 and the second in instructions 6–13 of Algorithm 2) with an additional pair of sampling and likelihood operations introduced in the first step, relatively to the SISR.

DOI: 10.1002/acs

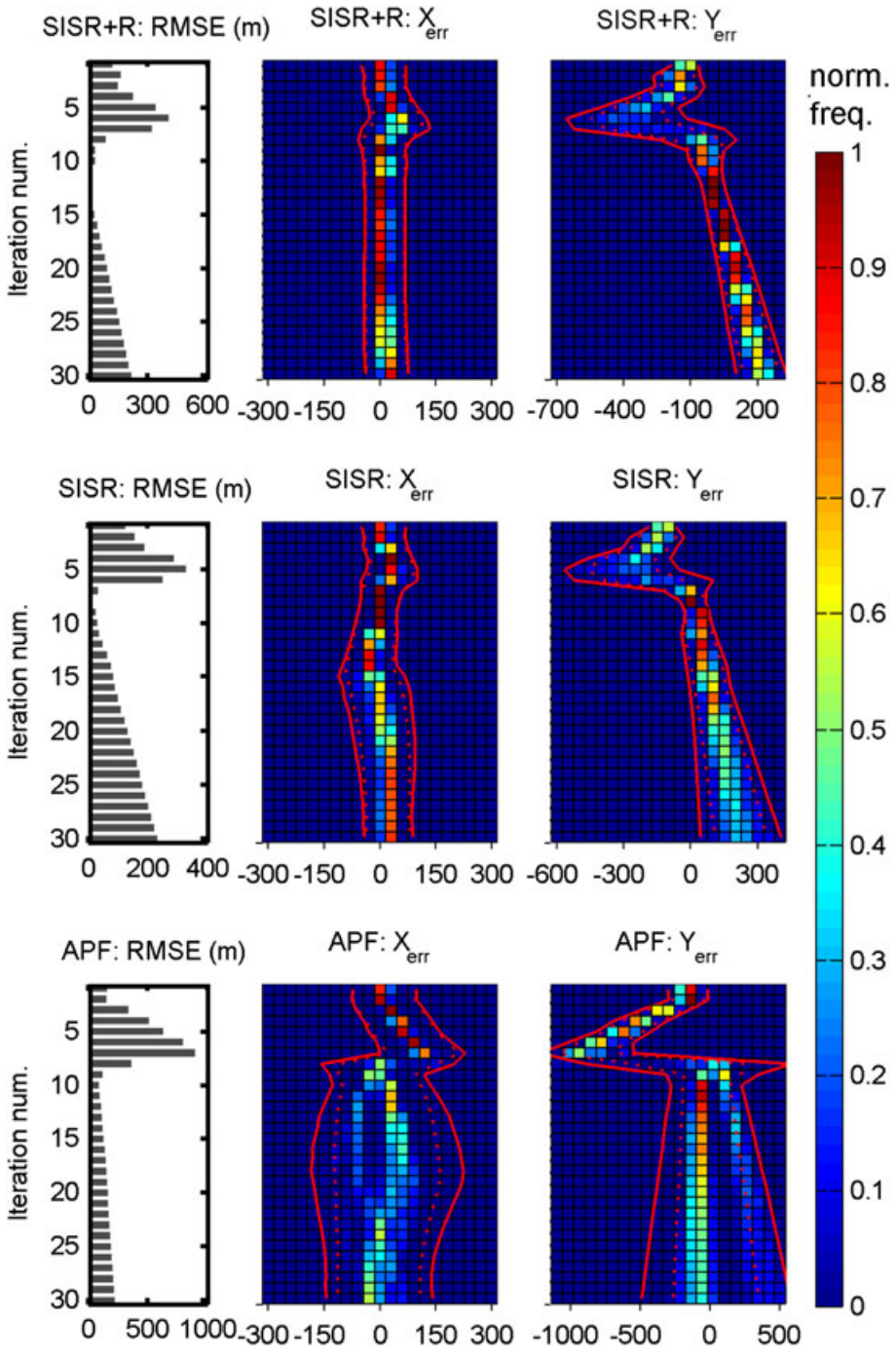

Figure 11. Approximate distributions of position estimation errors and corresponding $RMSE_t^{100}$ obtained with the sequential importance sampling + resampling (SISR+R), sequential importance sampling (SISR), and auxiliary particle filter (APF), applied to the bearings-only problem ($N = 5000$). The mean-centered 1σ and 2σ intervals are represented by the dotted and the continuous lines, respectively. [Colour figure can be viewed at wileyonlinelibrary.com]

The additional sampling operation justifies in part the considerably larger execution time of each AFP iteration, as recognized in [55]; notice also that the computation of measurement likelihoods (as in instructions 3 and 8 of the APF) is normally more demanding than random sampling and contributes to the slower execution of the filter. We note that the MPF, which is faster than the APF, also computes two measurement likelihoods (in instructions 5 and 9 of Algorithm 3) but in practice

DOI: 10.1002/acs

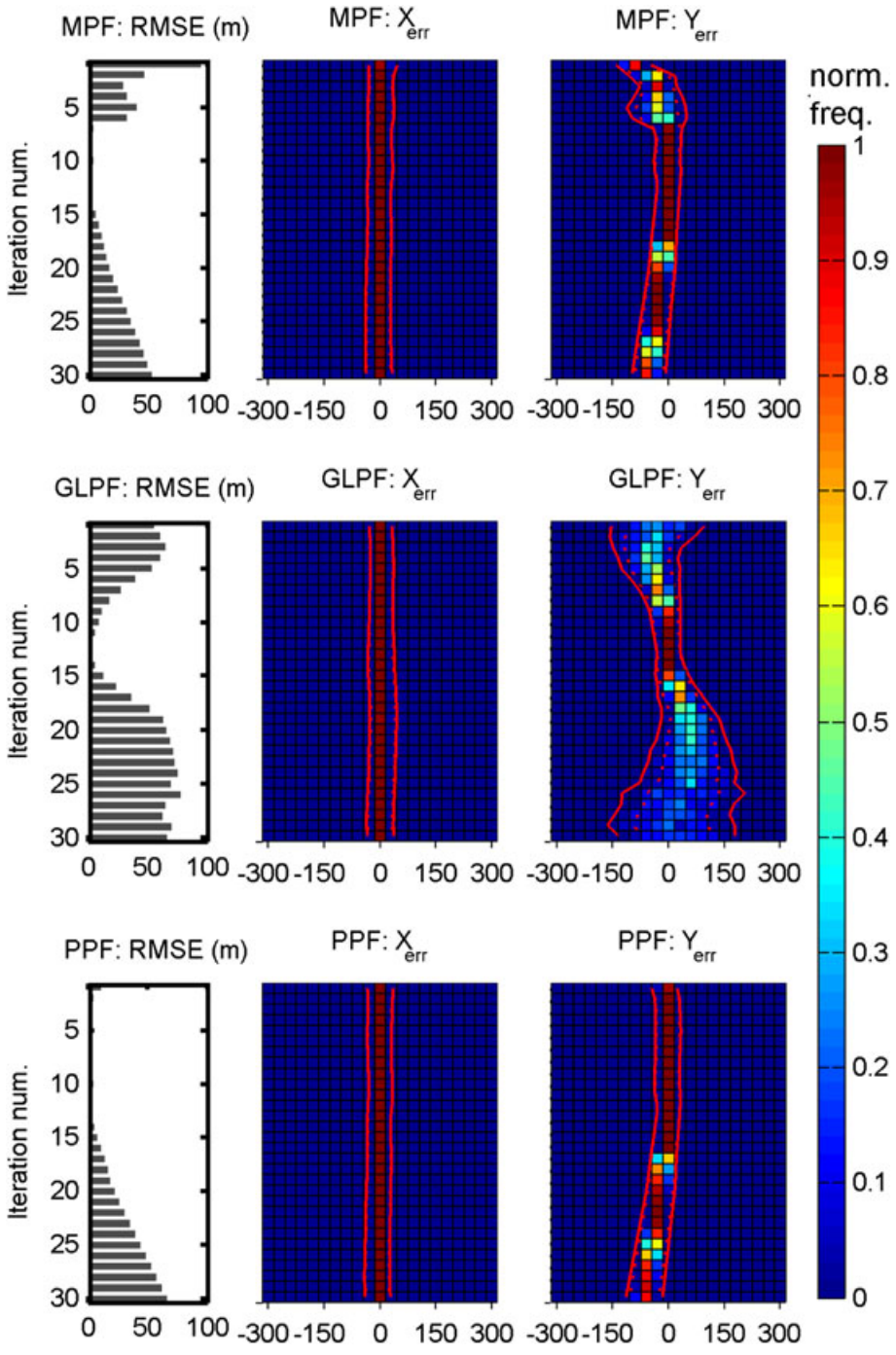

Figure 12. Approximate distributions of position estimation errors and corresponding $RMSE_t^{100}$ obtained with the mixture particle filter (MPF), generic likelihood particle filter (GLPF), and prior-correction particle filter (PPF), applied to the bearings-only problem ($N = 5000$). The mean-centered 1σ and 2σ intervals are represented by the dotted and the continuous lines, respectively. [Colour figure can be viewed at wileyonlinelibrary.com]

these are implemented in a single instruction because the operation is applied to the two components of a mixture of particles defined at a given instant of time. The amount of memory required by any of filters is very low and does not constitute a limiting factor, given the typical RAM capacity of modern personal computers.

DOI: 10.1002/acs

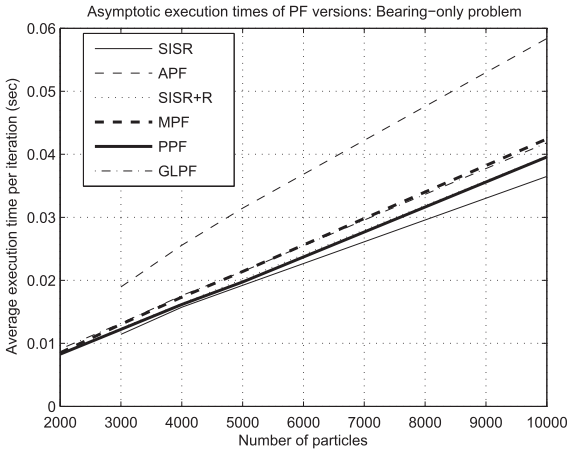

Figure 13. Comparison of CPU time per iteration used by the different filters in the bearings-only problem for different sample sizes.

Table II. RAM and CPU time per iteration used by particle filters in the Bearings-only problem using 5000 particles.

	SISR	SISR+R	APF	MPF	PPF	GLPF
RAM (kilobytes)	396.0	396.0	3276.0	400.0	360.0	1028.0
CPU (seconds)	0.0192	0.0202	0.0314	0.0214	0.0197	0.0215

APF, auxiliary particle filter; GLPF, generic likelihood particle filter; MPF, mixture particle filter; PPF, prior-correction particle filter; SISR, sequential importance sampling; SISR+R, sequential importance sampling + resampling.

6. SOLUTION OF THE TERRAIN-AIDED NAVIGATION PROBLEM USING PARTICLE FILTERS

In this section, we assess the relative performance of some of the aforementioned algorithms, applying them to the solution of the TAN problem. For filter benchmarking, we use simulated navigation data and a bathymetric map of a real underwater scenario, which combines patches of flat ocean floor interwoven with topographic ridges. This environment offers ideal conditions to test the robustness of TAN estimation algorithms because of the large variability of terrain gradients observed in the area and the high degree of terrain symmetry that characterizes the most prominent elevations. Based on the results of this analysis, in Section 6.3, we select the filter that provides the best navigation results to be applied to experimental data obtained with a robotic marine vehicle.

6.1. System modeling and estimation methods

6.1.1. Process and measurement models. In the models that follow, we assume the availability of accurate depth measurements combined with precise modeling of tide level variations. From these data, we obtain an accurate estimate of the absolute vertical position of the vehicle.[†] Thus, the variable z used in the models is considered an input (not a state variable) and the position and velocity estimation is reduced to a two-dimensional estimation problem. The corresponding discrete-time process model is

[†]It is important to notice that very precise modeling of tide level variations is not possible in general, particularly in the shelf and near-coastal zones. However, the assumption made in our work is justified taking in consideration the scenario where the simulated experiments occur and is supported by recent accuracy assessments of tide models; see e.g. Stammer, D., et al., *Accuracy assessment of global barotropic ocean tide models*. Reviews of Geophysics, 2014. 52: p. 243-282. We consider also relevant to mention that in the absence of accurate tide models it is possible to augment the navigation filters by one dimension in order to implement online estimation of depth biases; this is expected to increase navigation accuracy, particularly in poorly excited terrain.

DOI: 10.1002/acs

$$\mathbf{x}_{t+1} = F\mathbf{x}_t + G_{u,k}\mathbf{u}_t + L\zeta_t \tag{20}$$

where F and L are proper size matrices, $\mathbf{x} = [x, y, b_x, b_y]^\top$ is the state vector, and ζ_t represents the process noise sequence. The input coupling matrix $G_{u,k}$ is parameterized by the yaw angle ψ and the yaw rate r; the product $G_{u,k}\mathbf{u}_t$ implements an integrator that transforms the body-referenced velocities represented in the input vector $\mathbf{u}_t = [v_u, v_v, v_w, \psi, r, z]^\top$ to horizontal displacements in the inertial reference frame (Appendix-A in [36]). The following variables are used in the sequel: $\mathbf{p} = [x, y]^\top$ is the position of the vehicle, $V = [v_u, v_v]^\top$ represents its horizontal linear velocity relative to the water, supplied by the Doppler, expressed in $\{B\}$, and $\mathbf{b} = [b_x, b_y]^\top$, expressed in $\{I\}$, represents the velocity bias introduced by ocean currents that must be estimated when the DVL is operated in water-lock mode; this vector can also be used to represent the Doppler velocity biases when the unit operates in bottom-lock. Assuming additive measurement noise, the discrete-time observation model is

$$\mathbf{y}_t = h(\mathbf{x}_t) + \eta(\mathbf{p}_t) \tag{21}$$

where $\mathbf{y}_t$ is a vector representing the simultaneous measurements taken at each iteration and η represents the measurement errors as a function of the three-dimensional position $\mathbf{p}_t$. Detailed derivations of the models presented previously can be found in [36].

6.1.2. Noise models. The discrete-time process noise sequences represented in ζ_t are assumed mutually independent and Gaussian, with intensity noise represented by matrix Q. A realistic measurement model is used where the time-varying error is a function of the sonar beam-width, slant range, and local terrain gradient. The measurement noise variables represented in (21) by the vector η are considered mutually independent and are characterized by the time-varying measurement noise intensity matrix R_t; notice that this matrix is the same used in the computation of the Fisher information matrix in (11). Please refer to [36] for details of these models.

6.1.3. The Cramér-Rao bound for terrain-aided navigation. The TAN Cramér-Rao lower bound (CRLB) is the inverse of the TAN information matrix defined in paragraph 4.4.2, eq. (11), that is,

$$CRLB_t = J_t^{-1}.$$

We note, however, that simulations are executed in a Bayesian framework, as implemented in our prior work, [36]; the Bayesian or posterior CRLB (PCRLB) is evaluated according to the recursive formulas presented by Tichavsky in [73]. According to this formulation, the lower bound is a function of the terrain gradient $H_t^\top$ and the measurement noise intensity R_t, and the system explores an ensemble of random trajectories in the state space, centered at the nominal trajectory. The mathematical expectation involved in the expressions of the PCRLB is thus approximated by the average taken over the ensemble of state vector realizations or target trajectories generated by the Monte Carlo algorithm. The theoretical derivation of the Bayesian CRLB is presented in [74]; a comprehensive discussion of methods for its computation and their respective applications can be found in [75].

As the reader may notice, in the areas of uninformative terrain, the conditional expectation often does not reach the PCRLB. This issue is observed frequently in the estimates provided by terrain-based navigation algorithms and is inherent to particle filter implementations, which – as in our case – use synthetic process noise, which is not considered in the computation of the lower bound. We consider that, despite its relevance, a lengthy discussion of this topic is out of the scope of the manuscript. The interested readers may find a detailed analysis of this topic in our technical report [76], which is available online through this link.

6.1.4. Marginalized particle filter set-up. The nonlinear function $h(.)$ in the measurement model represented by (21) only uses part of the state vector $\mathbf{x}$, that is, $h(\mathbf{x}) = h(x, y)$. Because model (20) is linear in the states and the additive noise is Gaussian, the PF can be Rao-Blackwellized; see the seminal work [77]. This is achieved by decomposing the state vector into two parts: $\mathbf{x} = [\mathbf{x}^{pf}, \mathbf{x}^{kf}]^\top$,

DOI: 10.1002/acs

where $\mathbf{x}^{pf} = [x, y]^\top$ represents the part of the state vector estimated by the PF and $\mathbf{x}^{kf} = [b_x, b_y]^\top$ is the part of the state vector estimated by a Kalman Filter; the approach is described in [26, 78]. We decompose similarly the state transition matrix F, the input coupling matrix $G_{u,k}$, the noise coupling matrix L, the input and noise vectors $\mathbf{u}_t$ and ζ_t, and the discrete-time process noise intensity matrix Q. In the sequel, we use $\mathbf{u}_t = [v_{u_t}, v_{v_t}, \psi_t, r_t, z_t]^\top = [V_t, \psi_t, r_t, z_t]^\top$. The prediction and updating steps of the different filters are implemented according to the expressions introduced in Section 4. The Rao-Blackwellized formulation is applied to all the filters under analyses; the expressions used by the PPF are

Prediction

$$\mathbf{x}_{t+1}^{pf,i} = \mathbf{x}_t^{pf,i} + F^{pf}\mathbf{x}_t^{kf,i} + G_{u,k}^{pf}V_t + \zeta_t^{pf} \tag{22}$$

$$\mathbf{x}_{t+1}^{kf,i} = F^{kf}\left[\hat{\mathbf{x}}_{t|k-1}^{kf,i} + K_t v_t\right] \tag{23}$$

where K_t denotes the Kalman filter gain, $v_t = \mathbf{x}_{t+1}^{pf,i} - \mathbf{x}_t^{pf,i} - \left(F^{pf}\hat{\mathbf{x}}_{t|k-1}^{kf,i} + G_{u,k}^{pf}V_t\right)$, and

$$\zeta_t^{pf} \sim \mathcal{N}\left(0, F^{pf}P_{t|k-1}^{kf}\left(F^{pf}\right)^\top + L^{pf}Q^{pf}\left(L^{pf}\right)^\top\right). \tag{24}$$

Update

The formulation used in the update step depends on the importance function used by the filter that leads to distinct PF implementations. As shown in the previous sections, it may involve the prior $p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$, the likelihood function of the observations given the predicted states $p(\mathbf{y}_t|\mathbf{x}_t^i)$, or both. According to the PPF, the weights update expression is

$$w_t^i \propto w_{t-1}^i f_w\left(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i\right) h_v\left(\mathbf{y}_t|\mathbf{x}_t^i\right). \tag{25}$$

The set of importance weights $\left\{w_t^i\right\}_{i=1}^N$ used in the sequel is assumed to be normalized.

Point estimates

A point estimate of the current state $\hat{\mathbf{x}}_t$ and the associated covariance matrices P_t can be obtained using the normalized weights via the equations

$$\hat{\mathbf{x}}_t^{MMS} \simeq \sum_i^N \omega_t^i \mathbf{x}_t^i = \sum_i^N \omega_t^i \begin{bmatrix} \mathbf{x}_t^{pf,i} \\ \mathbf{x}_t^{kf,i} \end{bmatrix} \tag{26}$$

$$P_t^{pf} = \sum_i^N \omega_t^i \left(\mathbf{x}_t^{pf,i} - \hat{\mathbf{x}}_t^{pf,MMS}\right) \cdot \left(\mathbf{x}_t^{pf,i} - \hat{\mathbf{x}}_t^{pf,MMS}\right)^\top \tag{27}$$

$$P_t^{kf} = P_{t|k}^{kf} + \sum_i^N \omega_t^i \left(\hat{\mathbf{x}}_{t|k}^{kf,i} - \hat{\mathbf{x}}_t^{kf,MMS}\right)\left(\hat{\mathbf{x}}_{t|k}^{kf,i} - \hat{\mathbf{x}}_t^{kf,MMS}\right)^\top. \tag{28}$$

6.2. Terrain-aided navigation benchmarking in simulations

This section is dedicated to benchmarking in simulations the performance of particle filters applied to TAN. The study compares the results obtained with the SISR, MPF, and PPF filters that systematically produced consistent results in the solution of the TAN problem. The other versions mentioned in the paper diverged frequently in these tests, despite our best efforts to enhance their performance; we analyze this issue in Section 6.2.4.

DOI: 10.1002/acs

6.2.1. Navigation problem and assumptions. The navigation system under analysis uses the vehicle kinematics based on measurements supplied by a DVL and an attitude and heading reference unit to perform dead-reckoning. However, there are unknown ocean currents that introduce a velocity bias that must be estimated.

It is assumed that the vehicle is leveled horizontally and stabilized in roll and pitch, that is, $\phi = \theta = 0$ such that the angular velocity yaw rate is given by $r = \dot{\psi}$. Steady-state rejection of the rate gyro bias as well as rejection of the fluxgate high-frequency noise is achieved by complementary filtering; refer to [36] for details. The velocity vector of the oceanic current is considered constant or varying very slowly in time. Without loss of generality, it is assumed that the AUV navigates at constant depth, which can be measured by using a pressure cell.

6.2.2. General set-up of the simulations. The scenario used in the simulations is the *D. João de Castro* sea mount, a submerged volcanic summit in the Azores archipelago at coordinates $38.233°N$, $26.633°W$. The bathymetry of the area ranges from approximately 20 to 60 m with very steep gradients in some locations. The prior map of the terrain represents an area with approximately 800 m$\times$ 400 m and is represented by a matrix of altitude values gridded with 1-m spacing; see the contour map in Figure 2.

In the tests, the TAN algorithm is run along a lawn-mower trajectory that crosses areas of high terrain excitation, interwoven with areas of much smoother topography; refer to Figure 3. The filters were run in simulations using different sample sizes: $N = 500$, 1000, 2000, and 5000. It was found that using a number of particles larger than 500 (plus 100 extra particles when applicable) did not improve the performance; for this reason the results presented correspond to simulations using $N = 500$. The main parameters used in the simulations are presented in Tables III and IV.

6.2.3. Terrain-aided navigation sensor suite. The reference map used in the tests was built from bathymetric data acquired automatically by the autonomous surface vehicle *Delfim*, designed and operated by Instituto Superior Técnico [79]. The bathymetry was acquired using a narrow-beam mechanically scanned sonar integrated with an attitude unit for correction of the ASV oscillations. For the sake of realism of the tests, the configuration of navigation sensors used in the simulations, including the DVL and the attitude and heading reference unit, is the same that was used for data

Table III. Simulated scenario used for terrain-aided navigation particle filter benchmarking.

Parameter	Value	Units
Type of trajectory	lawn-mower	
Depth of vehicle operation	10	m
Oceanic current velocity vector(m/s)	$[-0.5; 0.5]$	m/sec
Speed of oceanic current	1.4	kt
Vehicle's velocity rel.water (max)	4.9	kt
Sonar effective beam-width, ϕ_s	1	deg

Table IV. Parameters used in terrain-aided navigation particle filter benchmarking.

Parameter	value	units
Filter update frequency: T^{-1}	4	Hz
DVL error std. dev. of $[v_u, v_v]'$: $\sigma_u = \sigma_v$	$0.2\% + 0.001$	m/sec
Std. dev. of velocity bias $[b_x, b_y]'$: $\sigma_{b_x} = \sigma_{b_y}$	10^{-6}	m/sec
Intensity of process synthetic noise: $diag(Q^{add})$	$[1, 1]^\top$	m
Number of standard particles: N	500	
Number of extra particles: N_{extra}	100	
Duration of simulation	425	sec
Number of Monte Carlo runs	100	

DVL, Doppler velocity logger.

DOI: 10.1002/acs

Table V. Specifications of sonar sensors used in terrain-aided navigation particle filter benchmarking.

Sensor type	Maker and Model	Operating Freq.	Max. range	Beam-width
Forward-looking sonar	Tritech Micron Echosounder DST	500 KHz	50 m	6° conical
Transversal scanning profiler	Tritech Super Seaking DFP	0.6 MHz or 1.1 MHz	80 m 40 m	2° conical 1° conical
Doppler velocity logger	Teledyne RDI WHN 1200	1.2 MHz	18 m	1.2° (4-beam)

acquisition with the ASV; the main equipment specifications are presented in Table V. Because of the characteristics of the Doppler and the large variations of the bathymetry observed in the scenario, it was not possible to ensure the continuous operation of the DVL in bottom-lock. This limitation justifies the implementation of dead-reckoning adopted in the simulations that uses Doppler velocity measurements relative to the water instead of the sea-bottom.

Besides the mechanically scanned sonar mentioned previously, the suite of acoustic sensors includes a narrow-beam forward-looking sonar to provide the altitude and two slant range measurements used by the TAN algorithm. This setup is justified by prior work where we analyzed the performance of TAN filters using different sensor configurations [36].

6.2.4. Discussion of simulations results. To illustrate the main differences in the evolution of the probability densities used by the standard PF and the new filters, we present in Figure 14 the three-dimensional plots of the bivariate posterior PDFs of the position estimates computed by the SISR and the PPF filter. The results of 100 MC simulations are presented in Figure 15. These simulations do not reveal any significant difference between the MPF and the PPF in terms of estimation errors and illustrate clearly the superior estimation performance obtained with the new filters relatively to the SISR version. Besides achieving considerably smaller position estimation errors, the new algorithms are also considerably more efficient and faster at estimating the velocity biases introduced by oceanic currents. As can be seen in Figure 15, the estimator of the velocity bias converges rapidly and its RMS error attains the Cramér-Rao lower bound for $t \geqslant 125$ s. This efficiency in terms of velocity estimation has an important impact on the navigation performance because after detection of the velocity bias, it is possible to navigate by dead-reckoning during relatively long periods even with a non-informative terrain. Hence, besides their characteristics of robustness relatively to terrain symmetries, the new navigation filters are also more robust relatively to the problem of poor terrain excitation.

The APF, GLPF, and LPF algorithms diverge frequently when applied to solve the terrain navigation problem used in our tests. A careful analysis of this behavior shows that the underlying cause of divergence is the main algorithmic principle of these filters that makes them particularly susceptible to the presence of ambiguous terrain information. All the three methods implement likelihood-based sampling (in the case of the LPF) or re-sampling (the case of the APF and GLPF) at the prediction stage of the filter and as a consequence tend to concentrate the particles in regions of the state space with high measurement likelihoods. This strategy is effective in problems where process noise is small and measurement likelihoods are represented by narrow, single-peaked PDFs; otherwise, it may degrade performance considerably. The technique is prone to fail when the process noise is intrinsically large or artificially increased (e.g., to avoid sample impoverishment), and the measurement space leads to ambiguous observations represented by very broad or multi-peaked measurement likelihoods. In order to avoid divergence under these conditions, the navigation filters must be highly robust to terrain ambiguity.

The operation of the likelihood-based algorithms (GLPF, LPF, and APF) is illustrated in Figure 16, which shows the clouds of particles generated in this scenario by these filters along the initial 33 s of simulation; these can be compared with the corresponding sequences of particles used by a prior-based filter, the PPF. As shown in these plots, the likelihood-based filters tend to distribute the particles along the depth isolines instead of the vehicle trajectory; this is in contrast to

DOI: 10.1002/acs

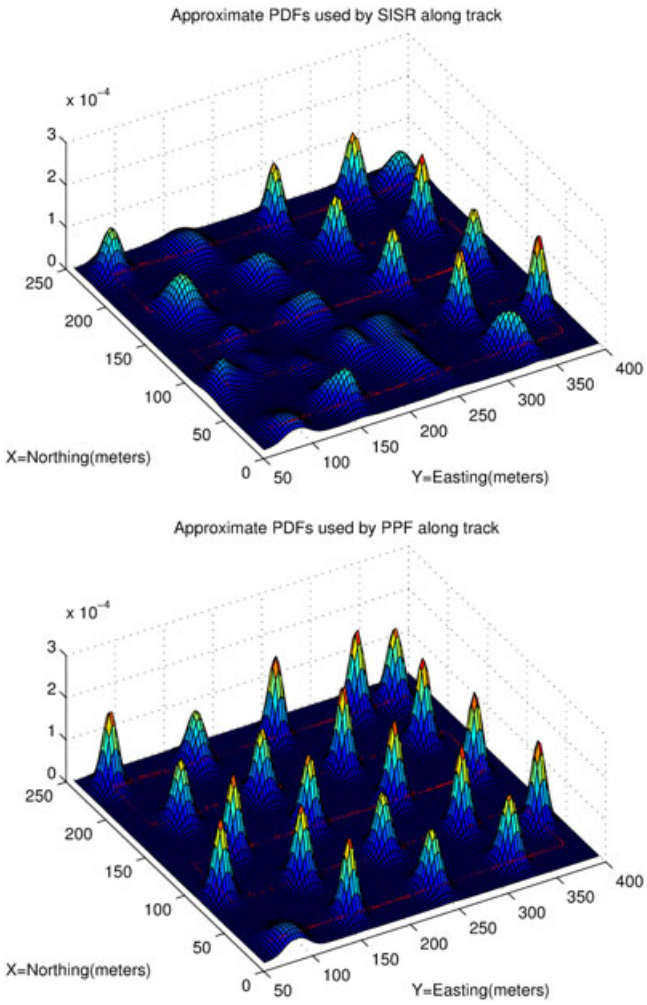

Figure 14. Evolution of the approximate posterior probability density functions (PDFs) used by the sequential importance sampling (SISR) (left) and the prior-correction particle filter (PPF) (right) along-track for the terrain-aided navigation problem. The red line underlying the PDF plots represents the true path followed by the vehicle. [Colour figure can be viewed at wileyonlinelibrary.com]

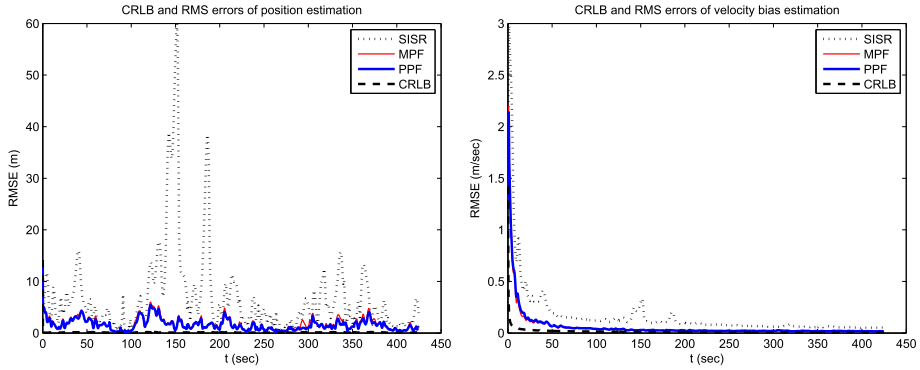

Figure 15. Root mean squared error (RMSE) of position (left) and velocity (right) obtained with different filters, compared with the posterior Cramér-Rao lower bound (PCRLB). [Colour figure can be viewed at wileyonlinelibrary.com]

DOI: 10.1002/acs

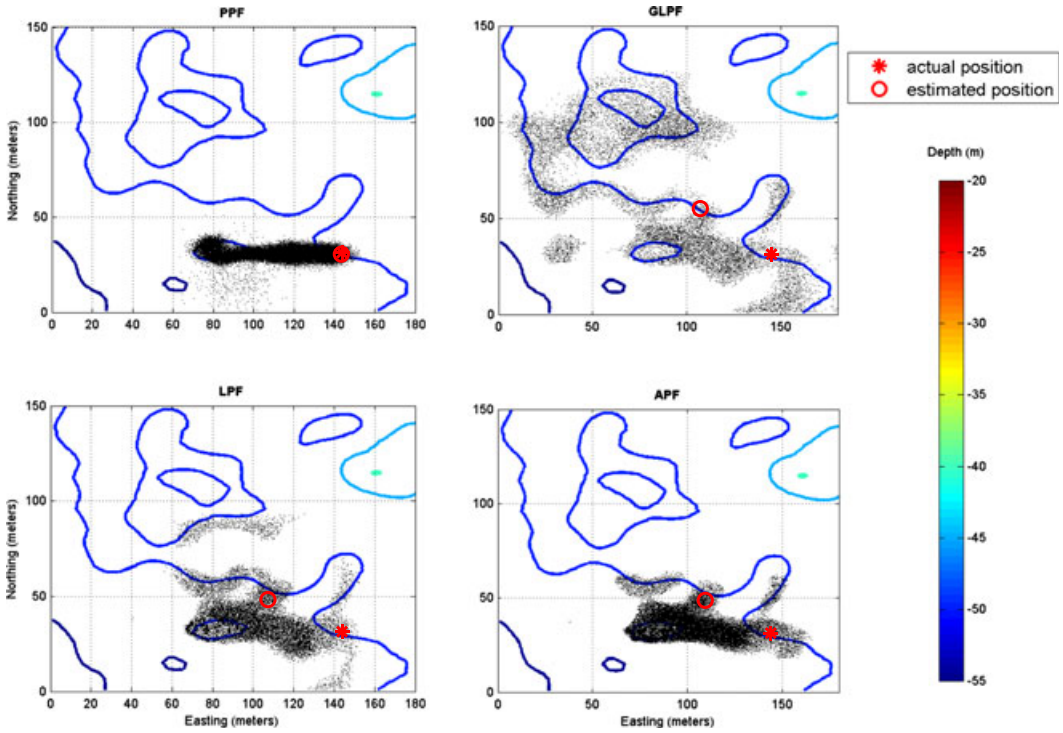

Figure 16. Clouds of particles generated in the terrain-aided navigation problem by the prior-correction particle filter (PPF) and the likelihood-based algorithms (generic likelihood particle filter (GLPF), likelihood particle filter (LPF), and auxiliary particle filter (APF)) along the initial 33 s of simulation. [Colour figure can be viewed at wileyonlinelibrary.com]

what happens with the PPF, which locates the cloud of particles around the true path. We notice that the auxiliary particle filter spreads the particles in better accordance with the trajectory predicted by the kinematic model than the other likelihood filters. This is an expected result because the APF combines sampling from an approximation of the prior with likelihood resampling at the prediction stage, while the GLPF and LPF filters do not use prior information at this stage.

These examples illustrate the typical behavior of the GLPF, LPF, and APF, which diverge in this phase of the simulation and are unable to recover. We conclude that one of the filters that we introduced, the GLPF, like the two classical likelihood-based particle filters, does not perform adequately when applied to the TAN problem. In contrast, we obtained accurate estimates of position with the other two algorithms proposed in the paper. These encouraging results motivated us to test the filters in practical experiments.

6.3. Terrain-aided navigation experiments with in-water collected data

One of the main objectives of this work is the implementation of terrain-aided navigation of AUVs using standard navigation sensors and relying on affordable sonar equipment for real-time acquisition of bathymetric data. For online acquisition of altitude data, it was proposed to use the range measurements provided by a Doppler velocity logger, which is a standard instrument used for dead-reckoning navigation in underwater robotics. The site chosen for the tests is a very shallow-water lake with an area of approximately 300 m $\times$ 200 m located at *Doca do Oceanário - Parque das Nações* ($38.765°N$, $9.093°W$), in Lisbon (Figure 17 (top)). The robotic platform used in the experimental tests is an autonomous vehicle of the class *Medusa* developed by the Institute for Systems and Robotics of Instituto Superior Técnico (ISR/IST), shown in Figure 17 (bottom). The *Medusas* can be configured as autonomous surface vehicles or autonomous underwater vehicles. The ASV configuration includes an external GPS antenna that can be used to track the vehicle using real-time kinematics GPS, thus providing ground-truth localization data with centimetric accuracy.

DOI: 10.1002/acs

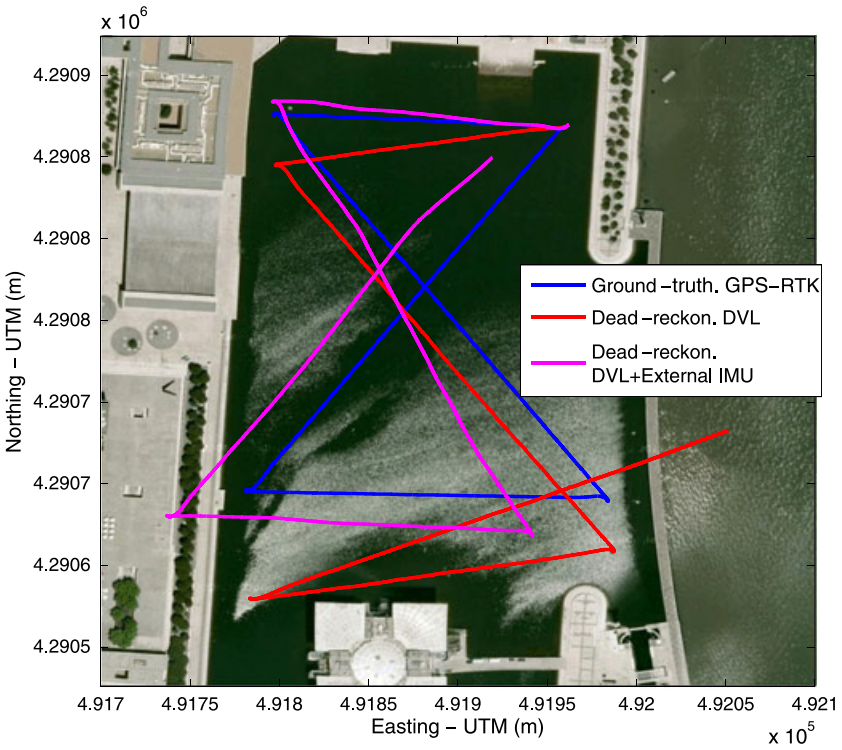

Figure 17. Top: true trajectory (blue) versus dead-reckoning estimates obtained with different motion sensor units. Bottom: the *Medusa* autonomous surface vehicle in the water at the test site in Lisbon. [Colour figure can be viewed at wileyonlinelibrary.com]

Table VI. Specifications of the Doppler velocity logger used in experimental trials.

Sensor function	Maker-Model	Frequency	Maximum range	Range accur.	Beam-width
Velocity sensor and altimeter	LinkQuest-NavQuest 600 MicroDVL	600 kHz	110 m	0.1	22° (4-beam)

DOI: 10.1002/acs

6.3.1. Configuration of the terrain-aided navigation experiments. The acoustic system used to acquire the altimetry data required by the TAN algorithms is a LinkQuest NavQuest 600 Micro Doppler velocity logger with a classical Janus four beam configuration. This DVL is a relatively inexpensive system, which is installed as a standard navigation instrument of the *Medusa* ASV. Besides providing velocity measurements used for dead-reckoning, the unit can acquire four depth measurements at each ping, with an approximate accuracy of 10 cm; the DVL is also equipped with a motion sensor, which provides relatively noisy measurements of the sensor attitude. An additional motion reference unit, which provides more accurate measurements of attitude and angular velocities, is installed on-board the vehicle: a VectorNav VN-100 inertial measurement unit and attitude heading reference system; see the equipment specifications in Tables VI and VII.

In the tests, a *Medusa* ASV, acting as a proxy to an AUV, was operated in fully autonomous mode, moved at the surface with a surge velocity of 0.5 m/s. The GPS real-time kinematics data acquired along-path by the vehicle is used in the present study exclusively to determine the navigation errors of terrain-aided navigation and dead-reckoning. The real path followed by the vehicle is shown in Figure 17, together with the dead-reckoning estimates based on DVL velocity measurements. In this experiment, the Doppler was used in a bottom-locking mode, thus providing measurement of the velocity with respect to the bottom.

The filter selected to execute the terrain navigation experiments is the PPF described in the previous sections. This selection is justified by the superior results obtained with this algorithm in the TAN benchmarking tests.

To build the prior map used by TAN, the bathymetry of the trial site was interpolated in a regular grid with 0.5 m spacing. As can be observed in the contour map represented in Figure 18, the topography of the site is very smooth except on the boundaries and in the Northern part of the lake, where a ramp that facilitates the access to the water is located.

Table VII. Specifications of motion sensors used in the experimental trials.

Sensor function	Maker-Model	Ang. rate noise roll, pitch, yaw	Head. accuracy Static, Dynamic	Roll, pitch acc. Static, Dynamic
Motion sensor	VectorNav VN-100	$0.0035°/s/\sqrt{Hz}$	2°, 2°	0.5°, 1°
DVL built-in motion sensor	LinkQuest NavQuest	not available (n.a.)	2°, n.a.	0.5°, n.a.

DVL, Doppler velocity logger.

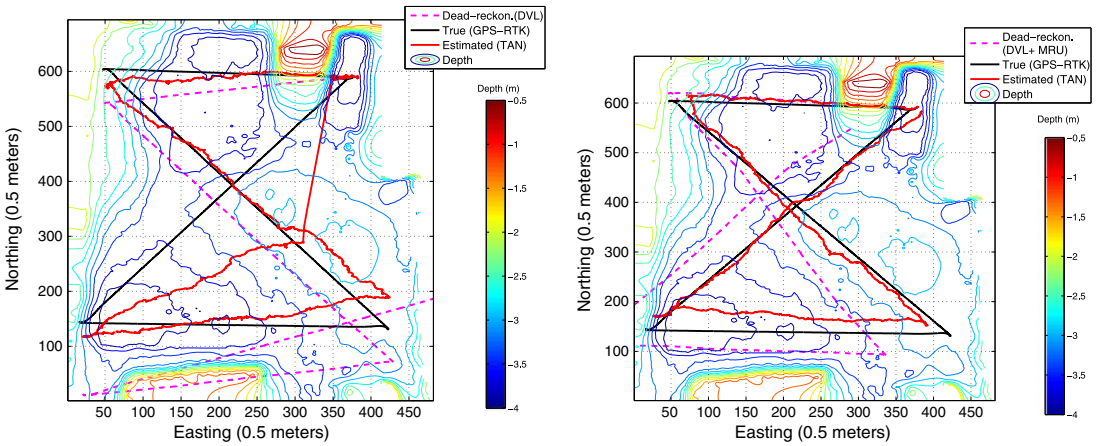

Figure 18. True trajectory, dead-reckoning implemented with Doppler velocity logger (DVL)-only (left), dead-reckoning with DVL+External inertial measurement unit (right), and corresponding terrain-aided navigation-estimated trajectories. [Colour figure can be viewed at wileyonlinelibrary.com]

DOI: 10.1002/acs

The analysis of the navigation data acquired in the trials revealed the poor accuracy of the DVL built-in motion sensor; in one of the legs, the heading measurements were biased in excess of 30 degrees. It was also observed that one of the DVL beams (beam 1) failed frequently and introduced large errors in the altitude measurements. For this reason, the utilization of this beam was inhibited in the TAN filter mechanization.

6.3.2. Presentation and discussion of experimental results. The tests discussed here include two versions of the navigation data: one that uses the attitude measurements acquired by the internal attitude unit of the DVL and another that uses the data obtained with the external motion sensor. In the analysis of the present results, it is worth noting that the orientation errors affect not only the estimation of the pose of the vehicle by dead-reckoning, but also the prediction of altitude measurements used by the TAN algorithm; these measurements are predicted by simulating the intersection of the acoustic beams with the bottom surface represented in the reference map. As such, heading errors can have a significant impact on the performance of TAN filters, especially in areas with large terrain gradients.

The trajectory estimated by the TAN filter based on the velocity and attitude data provided by the DVL is plotted in Figure 18 (left). In this figure, the true trajectory and the dead-reckoning estimate can be easily compared. The magnitudes of the TAN and dead-reckoning localization errors are plotted in Figure 19. Careful analysis of these plots show that the TAN filter diverged in the last leg of the test when the dead-reckoning error attains its maximum value because of a strong bias in the heading measurements provided by the internal attitude sensor of the DVL. In this part of the trajectory, the vehicle navigates over an almost flat terrain, which does not provide significant information for terrain navigation. However, the filter is able to recover when the vehicle reaches the area of high slope in the Northeastern part of the lake, converging approximately to the correct position and finishing the maneuver with a negligible error. This was due to the fact that the filter re-initialized after detecting its divergence, using the procedure mentioned in Section 2.1.2.

The best TAN results overall are obtained using the VectorNav motion sensor to complement the ground-referenced velocity data provided by the Doppler. As can be observed in Figure 18 (right), in this case the TAN estimates follow closely the true trajectory. The larger deviation occurs in the Southern leg where the topography does not convey enough information to enable the correction of the drift accumulated in the center of the lake. Figure 19 shows that the error decreases in the final part of the trajectory where the TAN filter explores efficiently the large terrain gradient observed in the area of the ramp.

The aforementioned presented results illustrate clearly the ability of TAN to achieve bounded localization errors even in poorly informative terrain and in the presence of faulty navigation sensors.

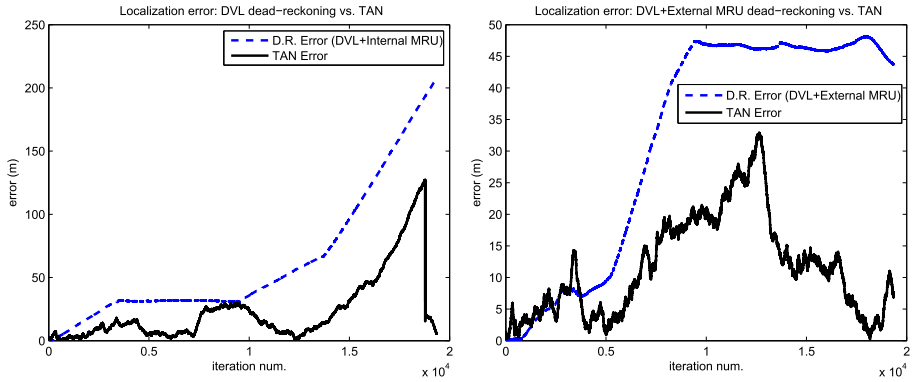

Figure 19. Absolute value of the localization error obtained with terrain-aided navigation (TAN) and dead-reckoning performed only with Doppler velocity logger (DVL) data (left), and performed with the DVL and the external MRU data (right). [Colour figure can be viewed at wileyonlinelibrary.com]

DOI: 10.1002/acs

7. CONCLUSION REMARKS AND GUIDELINES FOR FUTURE WORK

Motivated by the need to derive accurate and robust estimators with application to terrain-aided navigation of underwater robotic vehicles, the paper presented three novel particle filter algorithms and discussed their relative performance in comparison with other well-known filters. The new filters are the MPF, PPF, and GLPF. The classical algorithms used in the study are the SISR or bootstrap, APF, and LPF. In the tests where the SISR diverged systematically, a modified version usually known as SISR with 'roughening' (SISR+R) was used instead. In order to perform a comprehensive analysis of their asymptotic performance and determine the 'optimal' number of particles for each algorithm, all the filters were tested with several sample sizes. The tests addressed two nonlinear estimation problems: the univariate non-stationary growth model used frequently in econometrics and the bearings-only tracking, which is a well-studied problem in military and other engineering applications.

Among the filters applied to the solution of the one-dimensional problem, the MPF and the PPF present the best overall performance in terms of estimation accuracy and robustness to ambiguous measurements. The results obtained show that the classical bootstrap filter can provide an accuracy similar to those filters although at the expense of using a larger number of particles. The GLPF exhibits an estimation accuracy equal or superior to that obtained with classical PF versions such as the APF, SISR, and LPF.

The application of the new algorithms (PPF, MPF, and GLPF) to the solution of the bearings-only problem evidence the superior estimation accuracy achieved by these filters relatively to the other methods. The metrics of computational load and memory usage obtained in these tests show that all the filters present similar hardware requirements and can be executed in real-time using relatively inexpensive computers.

The performance achieved by the PPF and MPF algorithms in simulations using real bathymetric data demonstrated the advantage of using these filters in TAN applications. These methods can provide superior localization accuracy and increased robustness in the presence of topographic symmetries and flat terrain. The GLPF, APF, and LPF performed poorly in the TAN tests, diverging frequently in the presence of ambiguous terrain information. The good results obtained with the PPF in simulations were confirmed by its application with data acquired by a marine robotic vehicle in water trials using affordable, standard navigation sensors. The filter estimated the position of the vehicle along a path that crossed uninformative terrain during a significant part of the experiment and achieved a final positioning accuracy of a few meters in loop closure. To the best of our knowledge, this results constitutes a significant advance in terrain-aided navigation of small underwater vehicles.

Given the encouraging results obtained in our prior work, we are currently preparing the application of these filters to geophysical navigation of AUVs in more demanding experiments, including deeper water environments and tests of longer duration.

ACKNOWLEDGEMENTS

This research was supported in part by project TEBAS funded by the Korean Agency for Defense Development (ADD), project MORPH (EU FP7 under grant agreement No. 288704), and the FCT [PEst-OE/EEI/LA0009/2011]. The work of Pramod Maurya was partially supported by NIO, Goa, India.

REFERENCES

1. Barkby S, Williams SB, Pizarro O, Jakuba MV. A featureless approach to efficient bathymetric SLAM using distributed particle mapping. *Journal of Field Robotics* 2011; **28**(1):19–39.
2. Arulampalam S, Maskell S, Gordon NJ, Clapp T. A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* 2002; **50**(2):174–188.
3. Ristic B, Arulampalam S, Gordon NJ. *Beyond the Kalman Filter - Particle Filters for Tracking Applications*. Artech House, 2004.
4. Ekütekin V. *Navigation and control studies on Cruise missiles*. Middle East Technical University: PhD Thesis, 2007.

DOI: 10.1002/acs

5. Hollowell J. HELI/SITAN: A terrain referenced navigation algorithm for helicopters. *Position Location and Navigation Symposium, 1990. The 1990's - A Decade of Excellence in the Navigation Sciences. IEEE PLANS '90*, IEEE, Las Vegas, NV, USA, 1990; 616–625.
6. Bergman N, Ljung L, Gustafsson F. Terrain navigation using Bayesian statistics. *IEEE Control Systems Magazine* 1999; **19**(3):33–40.
7. Rice H, Mendelsohn L, Aarons R, Mazzola D. Next generation marine precise navigation system. *IEEE Position Location and Navigation Symposium. PLANS 2000*, PLANS IEEE Position, San Diego, CA, USA, 2000; 200–206.
8. Rice H, Kelmenson S, Mendelsohn L. Geophysical navigation technologies and applications. *IEEE Position Location and Navigation Symposium. PLANS'04*, Monterey, California, USA, 2004; 618–624.
9. Nygren I, Jansson M. Terrain navigation for underwater vehicles using the correlator method. *IEEE Journal of Oceanic Engineering* 2004; **29**(3):906–915.
10. Nygren I. Terrain navigation for underwater vehicles. *PhD Thesis*, Royal Institute of Technology, Sweden, 2005.
11. Carreno S, Wilson P, Ridao P, Petillot Y. A survey on terrain based navigation for AUVs. *OCEANS 2010*, IEEE, Seattle, WA, 2010; 1752–1758.
12. Melo J, Matos A. On the use of particle filters for terrain based navigation of sensor-limited AUVs. *OCEANS 2013 MTS/IEEE*, IEEE, Bergen, 2013; 294–301.
13. Jalving B, Mandt M, Hagen OK. Terrain referenced navigation of AUVs and submarines using multibeam echo sounders. *Undersea Defence Technology Conference - UDT Europe 2004*, Nice, France, 2004; 1–10.
14. Nygren I. Robust and efficient terrain navigation of underwater vehicles. *IEEE/ION Position, Location and Navigation Symposium (PLANS' 2008)*, IEEE, Monterey, CA, 2008; 923 –932.
15. Morice C, Veres S, Mcphail S. Terrain referencing for autonomous navigation of underwater vehicles. *Oceans 2009*, Europe, Bremen, 2009; 441–447.
16. Meduna DK, Rock SM, McEwen R. Low-cost terrain relative navigation for long-range AUVs. *OCEANS'2008. MTS/IEEE Oceans Conf.*, Quebec, CA, 2008; 1643–1649.
17. Meduna D, Rock S, Mcewen R. Closed-loop terrain relative navigation for AUVs with non-inertial grade navigation sensors. *Autonomous Underwater Vehicles (AUV), 2010, IEEE/OES*, Vol. 2010, Monterey, CA; 1–8.
18. Meduna D. Terrain relative navigation for sensor-limited systems with application to underwater vehicles. *PhD Thesis*, Stanford, 2011.
19. Anonsen KB, Hagen OK. An analysis of real-time terrain aided navigation results from a HUGIN AUV. *OCEANS 2010 IEEE-MTS Conference*, IEEE, Seattle, WA, 2010; 1431–1439.
20. Hagen OK, Anonsen KB, Mandt M. The HUGIN real-time terrain navigation system. *OCEANS 2010 IEEE-MTS Conference*, IEEE, Seattle, WA, 2010; 950–956.
21. Anonsen KB, Hagen OK, Hegrenaes O, Hagen PE. The HUGIN AUV terrain navigation module. *OCEANS 2013 IEEE-MTS Conference*, IEEE, San Diego, CA, 2013; 1966–1973.
22. Hagen OK, Anonsen KB, Saebo T. Low altitude AUV terrain navigation using an interferometric sidescan sonar. *OCEANS 2011*, IEEE, Waikoloa, HI, 2011; 2649–2656.
23. Donovan G. Position error correction for an autonomous underwater vehicle inertial navigation system (INS) using a particle filter. *IEEE Journal of Oceanic Engineering* 2012; **37**(3):431 –445.
24. Bergman N. Recursive Bayesian estimation - navigation and tracking applications. *PhD Thesis*, Linköping University, Sweden, 1999.
25. Gustafsson F, Gunnarsson F, Bergman N, Forssell U, Jansson J, Karlsson R, Nordlund P-J. Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing* 2002; **50**(2):425–437.
26. Nordlund PJ. Sequential Monte Carlo filters and integrated navigation. *Licenciate thesis*, Linköpings University, Sweden, 2002.
27. Frykman P. Applied particle filters in integrated aircraft navigation. *MSc thesis*, Linköpings Universitet, Sweden, 2003.
28. Shidan L, Liguo S, Xin L, Desheng W. Research on an improved terrain aided positioning model. *Journal of Software* 2011; **6**(5):937–943.
29. Nordlund PJ, Gustafsson F. Research on an improved terrain aided positioning model. *Technical Report*LiTH-ISY-R-2870, Linköpings Universitet: Sweden, 2008.
30. Anonsen KB, Hallingstad O. Terrain aided underwater navigation using point mass and particle filters. *IEEE/ION Position Location and Navigation Symposium*, San Diego, CA, 2006; 1027–1035.
31. Karlsson R, Gustafsson F, Karlsson T. Particle filtering and Cramér-Rao lower bound for underwater navigation. *2003 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, Hong-Kong, 2003; VI - 65–68.
32. Musso C, Oudjane N. Recent particle filter applied to terrain navigation. *Third International Conference on Information Fusion, 2000. FUSION 2000*, IEEE, Paris, France, 2000; 26–33.
33. Musso C, Oudjane N, Le Gland F. Improving regularised particle filters. In *Sequential Monte Carlo Methods in Practice*, Doucet A, de Freitas N, Gordon NJ (eds). Springer-Verlag: New York, 2001.
34. Casarin R. Online data processing: Comparison of Bayesian regularized particle filters. *Electronic Journal of Statistics* 2009; **3**:239–258.
35. Flament M, Fleury G, Davoust ME. Particle filter and Gaussian-mixture filter efficiency evaluation for terrain-aided navigation. *12th European Signal Processing Conference (EUSIPCO-2004)*, EURASIP, Vienna, Austria, 2004; 605–608.

DOI: 10.1002/acs

36. Teixeira FC. Terrain-aided navigation and geophysical navigation of autonomous underwater vehicles. *PhD Thesis*, Instituto Superior Técnico, Lisboa, 2007.
37. Murangira A, Musso C, Dahia K, Allard JM. Robust regularized particle filter for terrain navigation. *14th International Conference on Information Fusion (FUSION'2011)*, IEEE, Chicago, IL, 2011; 890–897.
38. Donovan G. Development and testing of a real-time terrain navigation method for AUVs. *OCEANS 2011*, IEEE, Waikoloa, HI, 2011; 420–428.
39. Murangira A, Musso C, Nikiforov I. Particle filter divergence monitoring with application to terrain navigation. *15th International Conference on Information Fusion (FUSION 2012)*, Vol. 2012, IEEE, Singapore, 2012; 794–801.
40. Zhang K, Li Y, Zhao J, Rizos C. A study of underwater terrain navigation based on the robust matching method. *Journal of Navigation* 2014; **67**(4):569–578.
41. Houts SE, Dektor SG, Rock SM. A robust framework for failure detection and recovery for terrain-relative navigation. *Unmanned Untethered Submersible Technology 2013*, Portsmouth, NH, 2013; 189–195.
42. Zhao B, Skjetne R, Blanke M, Dukan F. Particle filter for fault diagnosis and robust navigation of underwater robot. *IEEE Transactions on Control Systems Technology* 2014; **22**(6):2399–2407.
43. Teixeira FC, Quintas J, Pascoal A. AUV terrain-aided Doppler navigation using complementary filtering. *9th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC-2012)*, Arenzano, Italy, 2012; 313–318.
44. Liu JS, Chen R. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association* 1998; **93**(443):1032–1044.
45. Cappé O, Godsill SJ, Moulines E. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE* 2007; **95**(5):899–924.
46. Ho Y, Lee R. A Bayesian approach to problems in stochastic estimation and control. *IEEE Transactions on Automatic Control* 1964; **AC-9**(4):333–339.
47. Tanner MA. *Tools for Statistical Inference*, Springer series on statistics. Springer -Verlag: New York, 1996.
48. Chopin N. Central limit theorem for sequential Monte Carlo methods and its applications to Bayesian inference. *Annals of Statistics* 2004; **32**(6):2385–2411.
49. Ripley B. *Stochastic Simulation*. John Wiley and Sons: New York, 1987.
50. Tanizaki H. *Computational Methods in Statistics and Econometrics*. Marcel Dekker Inc.: New York, 2004.
51. Kong A, Liu JS, Wong WH. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association* 1994; **89**(425):278–288.
52. Gordon NJ, Salmond D, Smith AFM. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings - Radar and Signal Processing* 1993; **140**(2):107–113.
53. Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing* 2000; **10**(3):197–208.
54. Gustafsson F. Particle filter theory and practice with positioning applications. *IEEE A & E Systems Magazine* 2010; **25**(7):53–81.
55. Pitt M, Shephard N. Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association* 1999; **94**(446):590–599.
56. Hol JD, Schön TB, Gustafsson F. On resampling algorithms for particle filters. *Nonlinear Statistical Signal Processing Workshop*, Cambridge, United Kingdom, 2006; 79–82.
57. Yu JX, Tang YL, Chen XC, Zhao Q. An adaptive resampling strategy in particle filter. *Int Conf. on Wavelet Analysis and Pattern Recognition*, Guilin, 2011; 144–149.
58. Bachmann A, Williams S. Outlier handling when using particle filters in terrain-aided navigation. *IAV2004 - 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, 2004; 359–364.
59. Teixeira FC, Pascoal A, Maurya P. A novel particle filter formulation with application to terrain-aided navigation. *IFAC Workshop on Navigation Guidance and Control of Underwater Vehicles (NGCUV'2012)*, Porto, Portugal, 2012; 132–139.
60. Thrun S, Burgard W, Fox D. *Probabilistic Robotics*. MIT Press: Cambridge, MA, USA, 2005.
61. Thrun S, Fox D, Burgard W. Monte Carlo localization with mixture proposal distribution. *National Conference on Artificial Inteligence*, AAAI, Austin, TX, 2000; 859–865.
62. Papoulis A. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill: New York, 1965.
63. Paulino CD, Turkman MAA, Murteira B. *Estatstica Bayesiana*. Fundação Calouste Gulbenkian: Lisbon, 2003; 1386–1396.
64. Robert CP, Chopin N, Rousseau J. Harold Jeffreys's theory of probability revisited. *Statistical Science* 2009; **24**(2):141–172.
65. Kay SM. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall: Upper Saddle River, NJ, USA, 1993.
66. Kotecha J, Djuric PM. Gaussian particle filtering. *IEEE Transactions on Signal Processing* 2003; **51**(10):2592–2601.
67. Gilks WR, Berzuini C. Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society* 2001; **63 B**(1):127–146.
68. Heijden Fvd. Consistency checks for particle filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2006; **28**(1):140–145.
69. Hamilton JD. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica* 1989; **57**(2):357–384.
70. Bowman AW, Azzalini A. *Applied Smoothing Techniques for Data Analysis : the Kernel approach with S-Plus Illustrations*. Oxford University Press: New York, USA, 1997.

DOI: 10.1002/acs

71. Lin X, Kirubarajan T, Bar-Shalom Y, Maskell S. Comparison of EKF, pseudomeasurement, and particle filters for a bearing-only target tracking problem. In *Signal and Data Processing of Small Targets 2002*, vol. 4728, Drummond OE (ed.)., SPIE - International Society for Optics and Photonics: Bellingham, Washington, USA, 2002; 240–250.
72. Farina A. Target tracking with bearings-only measurements. *Signal Processing* 1999; **78**:61–78.
73. Tichavsky P, Muravchik CH, Nehorai A. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing* 1998; **46**(5).
74. Van Trees HL. *Detection, Estimation, and Modulation Theory*. John Wiley & Sons: New York, 1968.
75. Zuo L. Conditional Posterior Cramér-Rao Lower Bound and Distributed Target Tracking in Sensor Networks. *Phd Thesis*, Syracuse University, 2011.
76. Teixeira FC, Quintas J, Maurya P, Pascoal A. Novel particle filters with application to Terrain-Aided Navigation. *Technical ReportGEONAV-01-2016*, Instituto Superior Tecnico, Univ. of Lisbon, 2016.
77. Blackwell D. Conditional expectation and unbiased sequential estimation. *Annals of Mathematical Statistics* 1947; **18**(1):105–110.
78. Nordlund PJ, Gustafsson F. Marginalized particle filter for accurate and reliable terrain-aided navigation. *IEEE Transactions on Aerospace and Electronic Systems* 2009; **5**(4):1385–1399.
79. Pascoal A, Oliveira P, Silvestre C, Sebastião L, Rufino MC, Barroso V, Gomes J, Pascoal A, Oliveira P, Silvestre C, other. Robotic ocean vehicles for marine science applications: the Europen ASIMOV project. *OCEANS 2000 MTS/IEEE Conference and Exhibition*, Vol. 1, Providence, RI, USA, 2000; 409–415.

DOI: 10.1002/acs