A Box Regularized Particle Filter for state estimation with severely ambiguous and non-linear measurements
Nicolas Merlinge, Karim Dahia, Hélène Piet-Lahanier, James Brusey, Nadjim Horri

To cite this version:

HAL Id: hal-02294326
https://hal.archives-ouvertes.fr/hal-02294326
Submitted on 23 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A Box Regularized Particle Filter for state estimation with severely ambiguous and non-linear measurements

Nicolas Merlinge a,b Karim Dahia a Hélène Piet-Lahanier a James Brusey b Nadjim Horri b

ONERA—The French Aerospace Lab
(e-mail: first name . last name @onera.fr)

Coventry University, United Kingdom
(e-mail: first name . last name @coventry.ac.uk)

Abstract

The first stage in any control system is to be able to accurately estimate the system’s state. However, some types of measurements are ambiguous (non-injective) in terms of state. Existing algorithms for such problems, such as Monte Carlo methods, are computationally expensive or not robust to such ambiguity. We propose the Box Regularized Particle Filter (BRPF) to resolve these problems. Based on previous works on box particle filters, we present a more generic and accurate formulation of the algorithm, with two innovations: a generalized box resampling step and a kernel smoothing method, which is shown to be optimal in terms of Mean Integrated Square Error. Monte Carlo simulations demonstrate the efficiency of BRPF on a severely ambiguous and non-linear estimation problem, that of Terrain Aided Navigation. BRPF is compared to the Sequential Importance Resampling Particle Filter (SIR-PF), Monte Carlo Markov Chain (MCMC), and the original Box Particle Filter (BPF). The algorithm outperforms existing methods in terms of Root Mean Square Error (e.g., improvement up to 42% in geographical position estimation with respect to the BPF) for a large initial uncertainty. The BRPF reduces the computational load by 73% and 90% for SIR-PF and MCMC, respectively, with similar RMSE values. This work offers an accurate (in terms of RMSE) and robust (in terms of divergence rate) way to tackle state estimation from ambiguous measurements while requiring a significantly lower computational load than classic Monte Carlo and particle filtering methods.

1 Introduction

A state estimation problem is commonly formalized by: a dynamical state model, which represents the state evolution, and an observation model, which links the state with the measurements. These two models usually include uncertainty to account for unmodeled dynamics, perturbations and measurement errors. Two main frameworks can be used to model uncertainty, namely: probabilistic filters and set-based observers.

In the probabilistic scheme, uncertainty is modeled by a probability density function. The most commonly used probabilistic filter is the Kalman Filter, originally designed for linear problems (see [4] for recent Kalman Filter extensions). It was extended to non-linear problems through the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF). Kalman filters assume that the uncertainty probability distributions are Gaussian. In comparison, particle filter methods, which are based on Monte Carlo methods, handle non-gaussian and strongly non-linear systems. Ristic [21] provides a review of probabilistic state estimation techniques, from Kalman Filters to particle filters. In order to enhance the accuracy of particle filter estimation, a large variety of methods were proposed. In particular, the kernel smoothing regularization (Regularized Particle Filter [17]) was proposed to make the estimated density function fit the target density in an optimal way, based on the kernel theory of Silverman [23]. In what follows, kernel refers to a given probability density function. However, probabilistic techniques may diverge when strong measurement ambiguity arises. Measurement ambiguity refers to the case when the observation model is non-injective, i.e., when one measurement corresponds to several possible states.

In the set-based framework, uncertainty probability distributions are assumed to be unknown but bounded and are represented by bounded sets, as introduced by Schweppe [22]. Several set representations have been proposed, including: intervals and boxes [12], ellipsoids [15], and polytopes [20]. The output of those algorithms is thus not seen as a point estimate associated with an
uncertainty, as with probabilistic filters, but is considered as a bounded set that is guaranteed to contain the actual state. However, set-based techniques are often limited to linear or near linear systems, especially with regard to the observation model.

In order to produce an estimator able to handle strong non-linearities with minimal complexity, Gning [1] introduced the Box Particle Filter (BPF). The algorithm is similar to a conventional particle filter, except that each particle is a weighted box. Each measurement is also represented by a box. At each measurement time-step, the box particles are contracted, which means that they are replaced by a subset of themselves consistent with the measurements. The interval analysis framework [11] makes box manipulations locally simple, while the resulting union of all box particles can approximate a complex state density.

The BPF has the advantage of requiring significantly less computation than the classic Monte Carlo methods, for example the Sequential Importance Resampling Particle Filter (SIR-PF [9]) or Monte Carlo Markov Chains (MCMC [3]), for a similar estimation performance. Indeed, the SIR-PF requires a large number of point-wise (MCMC [3]) for a similar estimation performance. In-~

However, BPF still suffers some drawbacks. First, modelling a priori the boxes as uniform kernels leads to a loss of generality. Although this simple probabilistic assumption is of practical interest in the case of unknown uncertainty probability density functions, it excludes the case when other assumptions are available on those densities. This paper proposes a more general theoretical description of BPF. Nevertheless, BPF tends to be quite conservative for severely ambiguous problems. Indeed, the filter often yields excessively large confidence bounds around the estimate. An example of ambiguous measurements is Terrain Aided Navigation (TAN), for which BPF fails to accurately converge [16]. This is mostly due to the resampling step, which aims to avoid the degeneracy phenomenon. Resampling is triggered when too many box particles have a weight that is close to zero. They are considered inconsistent with respect to the measurements and are replaced by subdivisions of more highly weighted box particles. The subdivision is performed along one dimension of the state space and consists of a repaving of the box particle into a number of new box particles proportional to the original box particle weight. The total number of box particles remains unchanged. Although many problem-dependent subdivision approaches exist, there is no general subdivision method for BPF. In addition, the box resampling leads to an increasing number of box overlaps, which results in a lack of accuracy in terms of density approximation.

This paper first recalls the state estimation scheme (Section 2) and gives a generic formulation of BPF (Section 3). Section 4 presents the two main contributions that yield the Box Regularized Particle Filter (BRPF):

1. A new box particle subdivision approach for resampling, with an analytic solution based on an observability-linked criterion (Section 4.1). A significant accuracy enhancement compared to the conventional BPF is demonstrated by numerical simulations, in terms of Root Mean Square Error (RMSE) (Section 5).
2. The optimal smoothing of the posterior conditional density estimation using bounded kernel regularization expressed in terms of box parameters (Section 4.2). The optimal smoothing kernel is determined analytically by minimization of the Mean Integrated Square Error (MISE) criterion. The resulting performance enhancement compared to SIR-PF, MCMC, and the original BPF is demonstrated by simulation (Section 5).

2 Problem statement

The system evolution is modeled by a discrete dynamical model:

\[ x_k = f(x_{k-1}, u_k) + w_k, \]  

(1)

with the process noise \( w_k \in \mathbb{R}^d \) and a control input \( u_k \in \mathbb{R}^d \). For the sake of brevity, the control input will
be omitted. The observation equation is as follows:

\[
m_k = h(x_k) + v_k,
\]

where \( h \) is the observation model (potentially non-injective), and \( v_k \in \mathbb{R}^{d_v} \) is the measurement noise. Process and measurement uncertainties can be described using two different frameworks. The first approach consists of associating a probabilistic density function (pdf) to \( v_k \) and \( w_k \). However when the selection of a suitable structure for the pdf proves difficult for highly non-linear models, an alternative consists of representing these uncertainties using the unknown but bounded representation. It relies on providing only a description of the bounds within a value varies. These different descriptions lead to different estimation methods that are briefly recalled hereafter.

2.1 Probabilistic observers framework

In probabilistic schemes, the process noise and the measurement noise are associated with some probability density function: the transition density \( p(x_k | x_{k-1}) \) and the likelihood \( p(m_k | x_k) \).

The prediction step corresponds to the dynamical propagation of the conditional density. The conditional density is obtained by a convolution of the prior conditional density \( p(x_{k-1} | M_{k-1}) \) (where \( M_k = \{ \mathbf{m}_1, \ldots, \mathbf{m}_k \} \)) with the state transition density \( p(x_k | x_{k-1}) \), through the Chapman-Kolmogorov equation:

\[
p(x_k | M_{k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | M_{k-1}) dx_{k-1}.
\]

The correction step corresponds to the predicted conditional density \( p(x_k | M_{k-1}) \) updated by the measurements density \( p(m_k | x_k) \) and is obtained by the Bayes rule, under the assumption of statistically independent measurements:

\[
p(x_k | M_k) = \frac{p(m_k | x_k) p(x_k | M_{k-1})}{\int p(m_k | x_k) p(x_k | M_{k-1}) dx_k}.
\]

2.2 Set-based observers framework

In the set-based observer scheme, the process and measurement noises are modeled by bounded sets \( \{ w_k \} \) and \( \{ v_k \} \). Therefore, let the measurement set be defined as \( \{ m_k \} = \{ y \in \mathbb{R}^{d_m} \mid y - m_k \in \{ v_k \} \} \). Let \( \{ x_{k-1} \} \) be the previous state set estimation and \( \{ x_k \} \) the current state set estimation.

The prediction step corresponds to the propagation of all elements of the previous estimated set, plus the process uncertainty set:

\[
\{ x_{k|k-1} \} = \{ x \in \mathbb{R}^d \mid x - f(x') \in \{ w_k \} \forall x' \in \{ x_{k-1} \} \}.
\]

The correction step corresponds to the measurement update obtained by intersecting the predicted set:

\[
\{ x_k \} = \{ x \in \{ x_{k|k-1} \} \mid h(x) \in \{ m_k \} \}.
\]

Several methods have been introduced to describe these sets and their evolution. In this work, the algorithm developed relies on the use of specific set descriptions that are boxes and intervals. The interval framework is described hereafter.

2.3 Interval analysis

This section briefly recalls the interval analysis formalism described in Jaulin [11]. An interval \([ a \) of \( \mathbb{R} \) is defined as \([ a, b ] = \{ x \in \mathbb{R} \mid a \leq x \leq b \} \in \mathbb{IR} \) where \( \mathbb{IR} \) denotes the real interval space. Equivalently, it can be described by its center and its diameter as \([ a, b ] = (c, \delta) = \{ x \in \mathbb{R} \mid c - \delta \leq x \leq c + \delta \} \in \mathbb{IR} \).

A box is noted \([ a ] = [a, b] \in \mathbb{IR}^d \) with \( a \in \mathbb{IR}^d \) and \( b \in \mathbb{IR}^d \) the lower and upper bounds of \([ a ] \). It describes a hyperrectangle and can be written as a vector of intervals: \([ a ] = [a_1, b_1] \times \ldots \times [a_d, b_d] \in \mathbb{IR}^d \). Equivalently, it can be described by its center and its diameter \([ a ] = (c_{[a]}, \delta_{[a]}) \). Table 1 describes the most usual interval analysis operations. Note that the volume of a

<table>
<thead>
<tr>
<th>Description</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation ( \odot ) ((+, -, \times, /))</td>
<td>([ a ] \odot [ b ] = { x \in \mathbb{IR} \mid x \in a, y \in b } \mid]</td>
</tr>
<tr>
<td>Intersection</td>
<td>([ a ] \cap [ b ] = [\max(a, b), \min(a, b)]]</td>
</tr>
<tr>
<td>Diameter</td>
<td>(\delta_{[a]} =</td>
</tr>
<tr>
<td>By convention, (0)</td>
<td>([0] = \emptyset)</td>
</tr>
<tr>
<td>Volume of a box</td>
<td>(</td>
</tr>
<tr>
<td>Diameter of a box</td>
<td>(\delta_{[a]} =</td>
</tr>
<tr>
<td>Center of a box</td>
<td>(c_{[a]} = \frac{1}{2}(a + b)) ((\in \mathbb{IR}^d))</td>
</tr>
</tbody>
</table>

box corresponds to the Lebesgue measure. Functions of \( \mathbb{IR}^d \rightarrow \mathbb{IR}^d \) \((\delta_1, \delta_2 \in \mathbb{N}^*)\) have also to be adapted to interval framework. This yields the definition of inclusion functions [11]. An inclusion function of a function \( \psi \) from \( \mathbb{IR}^d \) to \( \mathbb{IR}^d \) is defined as \( \psi([a]) \) from \( \mathbb{IR}^d \) to \( \mathbb{IR}^d \). The output of \( \psi \) is a box \([\psi([a])] \in \mathbb{IR}^d \) that contains the output set \( \{ \psi([a]) \} \). In what follows, in order to limit the conservatism of box-based outer-approximations of output sets, inclusion functions
The predicted conditional density can be written as:
\[ p(x_k | M_{k-1}) = \sum_{i=1}^{N} w_{k-1} \pi_{k-1}^i \mathbf{1}_{[x_{k-1}]}(x_k), \]
where each box kernel \( \pi_{k-1}^i : \mathbb{R}^d \rightarrow \mathbb{R} \) is a known probability density function and satisfies \( \int_{[x_{k-1}]} \pi_{k-1}^i(x) dx = 1 \). The assumption is made that the dynamical model \( f : \mathbb{R}^d \rightarrow \mathbb{R}^d \) and the observation model \( h : \mathbb{R}^d \rightarrow \mathbb{R}^m \) are continuous. Initial box kernels \( \pi_0^i \) are also assumed to be continuous. Likewise, the measurement density \( \pi_m^i : \mathbb{R}^d \rightarrow \mathbb{R}^m \) is assumed to be continuous at each time-step. For the sake of brevity, kernel arguments are omitted in what follows. A computationally efficient solution is provided for the specific case of uniform kernels in Gning [1, 8].

### 3.1 Prediction step

Applying the Chapman-Kolmogorov equation (3) to the prior density (8) yields:
\[
p(x_k | M_{k-1}) = \int_{\mathbb{R}^d} p(x_k | x_{k-1}) \sum_{i=1}^{N} w_{k-1} \pi_{k-1}^i \mathbf{1}_{[x_{k-1}]}(x_k) dx_{k-1}
\]
\[
= \sum_{i=1}^{N} \left( w_{k-1} \int_{[x_{k-1}]} p(x_k | x_{k-1}) \pi_{k-1}^i \mathbf{1}_{[x_{k-1}]}(x_k) dx_{k-1} \right).
\]

Define the \( i \)th predicted kernel, whose support is included in box particle \([x_{k-1}^i]\):
\[
\pi_{k|k-1}^i(x_k) \triangleq \int_{[x_{k-1}^i]} p(x_k | x_{k-1}) \mathbf{1}_{[x_{k-1}]}(x_k) dx_{k-1}.
\]
The predicted conditional density can be written as:
\[
p(x_k | M_{k-1}) = \sum_{i=1}^{N} w_{k-1} \pi_{k|k-1}^i \mathbf{1}_{[x_{k|k-1}^i]},
\]
with
\[
[x_{k|k-1}^i] \triangleq [f([x_{k-1}^i])] + [w_k].
\]
As stated in Gning [8], for uniform kernels, the assumption can be made that:
\[
\pi_{k|k-1}^i = \int_{\mathbb{R}^d} u(x_k | x_{k-1}) p(x_k | x_{k-1}) dx_{k-1} \approx u(\{f([x_{k-1}^i])\} + [w_k]),
\]
3.2 Correction step

The correction step determines the posterior conditional distribution of the state with respect to the predictive distribution (11) and the measurement density. The measurement noise density \( \pi_m^i \) is assumed to be bounded by the box \([m_i]\). Bayes’ rule (4) leads to:
\[
p(x_k | M_k) = \frac{1}{q_k} \sum_{i=1}^{N} w_{k-1} \pi_{k|k-1}^i \mathbf{1}_{[x_{k|k-1}^i]} \pi_m^m \mathbf{1}_{[m_i]},
\]
where each box kernel \( \pi_{k|k-1}^i : \mathbb{R}^d \rightarrow \mathbb{R} \) is assumed to be continuous at each time-step. For the sake of brevity, kernel arguments are omitted in what follows. A computationally efficient solution is provided for the specific case of uniform kernels in Gning [1, 8].

However, the term \( \pi_{k|k-1}^m \pi_m^m \pi_{k|k-1}^i \) no longer sums to unity and is therefore not a pdf. Indeed, the support of the kernel \( \pi_{k|k-1}^i \mathbf{1}_{[x_{k|k-1}^i]} \) has been truncated by \([x_{k|k-1}^i] \subset [x_{k|k-1}]\) whose volume is lower or equal to that of \([x_{k|k-1}^i]\). Furthermore, it is multiplied by the measurement kernel \( \pi_m^m \), which leads to a new kernel proportional to \( \pi_{k|k-1}^m \pi_m^m \pi_{k|k-1}^i \). Therefore, it has to be normalised by \( \int_{[x_{k|k}^i]} \pi_{k|k-1}^m \pi_m^m \pi_{k|k-1}^i dx \), which yields,
\[
p(x_k | M_k) = \frac{1}{q_k} \sum_{i=1}^{N} w_{k-1} \int_{[x_{k|k}^i]} \pi_{k|k-1}^m \pi_m^m \pi_{k|k-1}^i dx \mathbf{1}_{[x_{k|k}^i]},
\]
where \( \pi_{k|k}^i \triangleq \frac{1}{q_k} \int_{[x_{k|k}^i]} \pi_{k|k-1}^m \pi_m^m \pi_{k|k-1}^i dx \) is the updated kernel. This kernel is only defined if \( \int_{[x_{k|k}^i]} \pi_{k|k-1}^m \pi_m^m \pi_{k|k-1}^i dx \neq 0,
i.e. if box $i$ is consistent with the measurement density. Else, it can be set by convention to $w^i_k = 0$.

Finally, the posterior conditional density is obtained:

$$p(x_k|M_k) = \sum_{i=1}^{N} w^i_k \pi^i_k 1_{[x^i_k]}, \quad (18)$$

where the update of weights is, for $k \geq 1$:

$$w^i_k \triangleq \frac{1}{q_k} w^i_{k-1} \int_{[x^i_k]} \pi^m_k (m_k - h(x)) \pi^i_{k|k-1}(x) dx. \quad (19)$$

The integral term in (19) can be interpreted as a consistency term between the box kernel $\pi^i_{k|k-1}(x)$ whose support is restricted to box $[x^i_k]$ and the measurement kernel $\pi^m_k (m_k - h(x))$. This integral term belongs to interval $[0, 1]$. The normalisation quotient is:

$$q_k = \sum_{i=1}^{N} w^i_{k-1} \int_{[x^i_k]} \pi^m_k \pi^i_{k|k-1} dx. \quad (20)$$

In the case of uniform kernels and a uniform measurement noise [8], the updated weights are obtained by:

$$w^i_k = \frac{1}{q_k} \frac{|[x^i_k]|}{|[x^i_{k|-1}|]} w^i_{k-1}. \quad (21)$$

### 3.3 Resampling step

As in a conventional particle filter, a resampling step is added to avoid the degeneracy phenomenon, when only a small number of box particles are consistent with the measurements and all others have a zero weight. When the resampling step is triggered, each box particle is duplicated in $n^i$ instances. If $n^i$ is zero, then the box particle is deleted. The total number of box particles usually remains the same after resampling, i.e., $\sum_i n^i = N$. The most common method used to determine $n^i$ is the multinomial resampling. The duplication number per box particle $n^i$ relies on the box particle weight, such that high weighted box particles are more likely to be duplicated than low weighted box particles.

A survey on resampling techniques for box particles filters can be found in Li [13]. The resampling step is not processed at every time-step and is triggered by a threshold based on the box particle weights. The most commonly used is the $N_{\text{effective}}$ criterion:

$$\frac{1}{\sum_{i=1}^{N} w^i_k} < \theta_{\text{eff}} N, \quad (22)$$

with $\theta_{\text{eff}} \in [0, 1]$ a normalized tuning threshold. This criterion is designed to reflect the case where the number of box particles with a low weight value exceed a given threshold.

The resampling step can be derived for the BPF scheme. Instead of duplicating the box particles, the BPF subdivides them along one dimension $d^i_{k|k}$ to increase the resolution of the state density estimation. Several strategies were proposed to choose the $d^i_{k|k}$ dimension along which to subdivide the current box particle. However, to the best of the author’s knowledge, they are either limited to a specific state representation, e.g. [2, 16, 5], fully observable states [14] or are not deterministic [7].

### 3.4 State estimation

A state estimate $\hat{x}_k$ can be derived from the box particles cloud such that:

$$\hat{x}_k \triangleq E[x_k | p(x_k|M_k)] \approx \sum_{i=1}^{N} w^i_{k-1} c^i_k. \quad (23)$$

For the sake of brevity, the box particles centers are denoted $c^i_k \triangleq E[x^i_k]$. An empirical covariance matrix can be defined by:

$$\hat{P}_k = \text{Cov}(x_k | p(x_k|M_k)) \approx \sum_{i=1}^{N} w^i_k (c^i_k - \hat{x}_k)(c^i_k - \hat{x}_k)^T. \quad (24)$$

The next section focuses on the resampling step. Section 4.1 introduces a generic box particle subdivision method. Section 4.2 presents a posterior density smoothing step called Regularization.

### 4 Box Regularized Particle Filter (BRPF)

In the previous section, a general probabilistic formulation of BPF was provided. However, as mentioned in 3.3, the box-resampling dimension subdivision does not have a generic definition in the literature and is often problem dependent. In this section, a generic box subdivision technique is described, based on variance evaluation.

Moreover, the resulting box particle cloud does not approximate the actual posterior conditional density in an optimal way with respect to the Mean Integrated Square Error (MISE) metric. Therefore, a kernel smoothing step is added after each resampling step to obtain an optimal box particles breakdown with respect to the MISE. We call the derived algorithm the Box Regularized Particle Filter.

#### 4.1 Box particles subdivision for resampling

The BPF box-resampling method aims to enhance the resolution of the state space exploration in its most likely
regions by replacing low-weighted boxes by subdivisions of high-weighted ones. For each box particle, the resampling algorithm provides an integer value indicating how many subdivisions will be performed. In this section, we consider a box \( i \) which has to be subdivided in \( n^i \) sub-boxes. A common approach is subdividing each box particle \( i \) along a single state dimension \( d^\text{cut},i \in \{1, \ldots, d\} \), where \( d \) is the number of state variables. Various methods were proposed in the literature. However, they are often too application-specific, not deterministic, or limited to fully observed states.

The objective of this section is providing a generic and deterministic formulation for the subdivision dimension \( d^\text{cut},i \) for each box particle. Our approach relies on an edge normalization by a lowest expected variance in the sense of the maximum likelihood estimator.

**Proposition 1** During the box subdivision resampling, each box can be subdivided along the edge that is the most pessimistic compared to a lowest expected variance. Provided that:

- The measurement noise (associated with the likelihood density) has a covariance \( R_k \in \mathbb{R}^{d_m \times d_m} \).
- The likelihood density has a single global maximum inside each box \( [x^i_k] \).
- The observation model \( h \) is differentiable.

The subdivision dimension \( d^\text{cut} \) is chosen by picking the largest coefficient of a normalised box particle’s diameter \( \delta^i_k \) in \([\delta^1_k, \ldots, \delta^d_k]^T\):

\[
d^\text{cut},i = \arg\max_{j \in \{1, \ldots, d\}} (\delta^i_j). \quad (25)
\]

The normalised diameter is computed from the box particle’s prior diameter \( \delta^i_k \equiv \delta_{[x^i]} \in \mathbb{R}^d \) and the inverse of the square root (e.g., Cholesky decomposition) of the lowest theoretical covariance \( \Sigma_k \):

\[
\tilde{\delta}^i_k = \left( \sqrt{\Sigma_k} \right)^{-1} \delta^i_k \in \mathbb{R}^d, \quad (26)
\]

where \( \Sigma_k \in \mathbb{R}^{d \times d} \) can be evaluated by:

\[
\Sigma_k = \hat{P}_k + \frac{1}{2} \sum_{i=1}^{N} w^i_k \left( \Sigma^i_k + V^i_k \right), \quad (27)
\]

where

\[
\Sigma^i_k = \text{Vec}^{-1} \left( \lambda T^i_k \otimes T^i_k + (1 - \lambda) I_d \right)^{-1},
\]

\[
\text{Vec} \left( \lambda H^i_k \otimes R_k H^i_k + (1 - \lambda) \Delta^i_k \right).
\]

and \( \Delta^i_k \) is a diagonal matrix whose diagonal terms are the squared box’s diameters \( d^i_k \) \((j \in \{1, \ldots, d\})\). The box can be subdivided in \( n^i \) new boxes along the dimension that maximises the normalised diameter:

\[
d^\text{cut} = \arg\max_{j \in \{1, \ldots, d\}} (\delta^i_j). \quad (29)
\]

**Proof:** In the following, the analytic expression of \( \Sigma_k \) is derived, leading to equation (27).

The measurement density \( p(m_k|x_k) = \pi^m_k (m_k - h(x_k)) \) is assumed to have a single global maximum \( \hat{x}^i_k \) inside each subset \([x^i_k] \), which is often the case in practice. \( \hat{x}^i_k \) satisfies:

\[
\hat{x}^i_k = \arg\max_{x \in [x^i_k]} p(m_k|x_k).
\]

Thus, one can link \( h(\hat{x}^i_k) \) and \( m_k \) as \( h(\hat{x}^i_k) = m_k \). The measurement variance \( \text{Cov}[m_k] = R_k \in \mathbb{R}^{d_m \times d_m} \) is assumed to be known. As a result, the maximum likelihood satisfies, for all \( i \):

\[
\text{Cov} \left[ h(\hat{x}^i_k) \right] = R_k, \quad (31)
\]

where the measurement covariance \( R_k \) does not depend on \( i \). On the other hand, the observation function \( h \) can be locally linearised to first order as follows:

\[
h(\hat{x}^i_k) = h(x^i_k) + H^i_k (\hat{x}^i_k - x^i_k) + o(\hat{x}^i_k - x^i_k), \quad (32)
\]

with \( H^i_k \equiv \frac{\partial h}{\partial x_{[x^i_k]}} \in \mathbb{R}^{d_m \times d} \) and \( x^i_k = E [x^i_k] \). This implies that:

\[
h(\hat{x}^i_k) - h(x^i_k) \approx H^i_k (\hat{x}^i_k - x^i_k), \quad (33)
\]

and that

\[
\left( h(\hat{x}^i_k) - h(x^i_k) \right) \left( h(\hat{x}^i_k) - h(x^i_k) \right)^T \approx H^i_k (\hat{x}^i_k - x^i_k) (\hat{x}^i_k - x^i_k)^T H^i_k. \quad (34)
\]

Thus, by taking the expected value of both sides of (34):

\[
E \left[ \left( h(\hat{x}^i_k) - h(x^i_k) \right) \left( h(\hat{x}^i_k) - h(x^i_k) \right)^T \right] \approx E \left[ H^i_k (\hat{x}^i_k - x^i_k) (\hat{x}^i_k - x^i_k)^T H^i_k \right] = H^i_k E \left[ (\hat{x}^i_k - x^i_k) (\hat{x}^i_k - x^i_k)^T \right] H^i_k = H^i_k \text{Cov}[\hat{x}^i_k] H^i_k. \quad (35)
\]

Thus, the local maximum likelihood must satisfy:

\[
\text{Cov} \left[ h(\hat{x}^i_k) \right] \approx H^i_k \Sigma^i_k H^i_k. \quad (36)
\]
with \( \Sigma^i_k \triangleq \text{Cov}[\hat{x}^i_k] \). Therefore, by combining (31) and (36), one can write:

\[
H^i_k \Sigma^i_k H^i_k^T \approx R_k. \tag{37}
\]

In practice, since the actual state \( x_k \) is unknown, \( H^i_k \) can be evaluated from \( H^i_k \approx \frac{\partial}{\partial c^i_k} |_{c^i_k = c^i_k} \) with \( c^i_k \) the center of the \( i^{th} \) box particle. Equation (37) imposes a constraint on \( \Sigma^i_k \) that depends on the observation equation.

However, if the rank of \( R_k \) is less than the rank of \( \Sigma^i_k \), some additional information needs to be added. Indeed, (37) only affects the coefficients of \( \Sigma^i_k \) that explicitly depend on the measurement in the observation function \( h \). In order to calculate the other coefficients, which are linked together through the dynamical model \( f() \), a possible solution is to introduce dynamical information with \( \Delta^i_k \triangleq \text{Diag}(\delta^2_k) \). \( \text{Diag}() \) transforms a vector in \( \mathbb{R}^n \) into a diagonal matrix in \( \mathbb{R}^{n \times n} \). Such techniques are known as regularisation techniques [8]. Therefore, the problem to solve is a trade-off between the observation constraint and the state constraint:

\[
\begin{align*}
\left\{ \left. \Sigma^i_k \right| & \text{argmin} (J(\Sigma)) \\
J(\Sigma) &= \lambda \left\| H^i_k \Sigma^i_k H^i_k^T - R_k \right\|_F^2 + (1 - \lambda) \left\| \Sigma - \Delta^i_k \right\|_F^2 \quad \Sigma \geq 0,
\end{align*}
\tag{38}
\]

where \( \lambda \in (0, 1) \) is a tuning coefficient. Criterion \( J \) can be minimized by first finding its derivative:

\[
\frac{\partial J}{\partial \Sigma} = \lambda tr \left( \left( H^i_k \Sigma H^i_k^T - R_k \right)^T (H^i_k \Sigma H^i_k^T - R_k) \right) + (1 - \lambda) tr \left( (\Sigma - \Delta^i_k)^T (\Sigma - \Delta^i_k) \right)
= 2\lambda \frac{\partial}{\partial \Sigma} tr \left( (H^i_k \Sigma H^i_k^T)^T (H^i_k \Sigma H^i_k^T) \right)
- 2\lambda \frac{\partial}{\partial \Sigma} tr \left( H^i_k \Sigma H^i_k^T R_k \right)
+ (1 - \lambda) \frac{\partial}{\partial \Sigma} tr (\Sigma^2)
- 2(1 - \lambda) \frac{\partial}{\partial \Sigma} tr (\Sigma \Delta^i_k)
= 2\lambda \left( H^i_k^T H^i_k \Sigma H^i_k^T H^i_k \right) - 2\lambda \left( H^i_k^T R_k H^i_k \right)
+ 2(1 - \lambda) \Sigma - 2(1 - \lambda) \Delta^i_k. \tag{39}
\]

where \( \frac{\partial J}{\partial \Sigma} = 0 \) is equivalent to:

\[
\lambda T^i_k \Sigma T^i_k + (1 - \lambda) \Sigma = \Omega^i_k, \tag{40}
\]

with \( T^i_k = H^i_k^T H^i_k \) and \( \Omega^i_k \triangleq \lambda H^i_k^T R_k H^i_k + (1 - \lambda) \Delta^i_k \). Using the Kronecker product properties, denoted \( \otimes \), derive (40) as follows:

\[
(\lambda T^i_k \otimes T^i_k + (1 - \lambda) I_d) \text{Vec}(\Sigma) = \text{Vec}(\Omega^i_k), \tag{41}
\]

where \( \text{Vec}() : \mathbb{R}^{d \times d} \to \mathbb{R}^{d^2} \) stands for the column-wise concatenation of a matrix.

As a result, the solution \( \Sigma^i_k \) can be obtained by:

\[
\Sigma^i_k = \text{Vec}^{-1} \left( (\lambda T^i_k \otimes T^i_k + (1 - \lambda) I_d) \text{Vec}(\Omega^i_k) \right), \tag{42}
\]

where \( \text{Vec}^{-1}(\cdot) : \mathbb{R}^{d^2} \to \mathbb{R}^{d \times d} \) gives a \( d \times d \) matrix representation of a \( d^2 \) vector whose elements are taken column-wise.

However, the resulting \( \Sigma^i_k \) matrix might not be positive definite. It can then be approximated by the nearest positive definite matrix, in terms of Frobenius norm, by \( \Sigma^i_k = \frac{1}{2} (\Sigma^i_k + V^i_k) \), where \( V^i_k \in \mathbb{R}^{d \times d} \) is obtained from the polar decomposition of \( \Sigma^i_k \), i.e. \( \Sigma^i_k = U^i_k V^i_k \) with \( U^i_k^T U^i_k = I_d \) (see Higham [10]).

The Maximum Likelihood covariance can therefore be approached by:

\[
\Sigma_k \triangleq \hat{P}_k + \sum_{i=1}^{N} w^i_k \Sigma^i_k. \tag{43}
\]

Each box diameter is normalised by the square root of \( \Sigma_k \) using (26) and the choice of \( d^{\text{cut}} \) is done using (29).

4.2 Bounded Kernel Smoothing by Regularization

In the previous section, a deterministic way to subdivide box particles was described. However, whatever be the subdivision method, this operation ends with a high correlation between the box particles parameters (centers and diameters). As a result, several box particles may exactly overlap and the posterior density \( p(x_k|M_k) \) is often poorly approximated, which leads to a biased state estimate. A possible solution to enhance the posterior density approximation is smoothing the box particles parameters distributions by adding stochastic bounded noise. This process, called kernel regularization, was presented in Musso [17] to improve the accuracy of conventional particles filters. It is based on the theory of Silverman [23] on kernel smoothing and density estimation. A kernel refers to a given probability density function, as introduced in Section 1. We introduced a first version of this approach for the Box Particle Filter in Merlino [16]. However, this formulation was performed by approximating each box particle by a uniform expectancy, which corresponds to the center of the box. As a result, the regularization formulation was equivalent to a conventional particle filter regularization.

The objective of this section is to determine an optimal smoothing kernel applied to the box parameters, in terms of the Mean Integrated Square Error (MISE) criterion defined in (48). Recall that the regularization is only performed when a resampling step is triggered by condition (22). We propose a new adaptation of the kernel.
regularization method, which relies on the whole boxes description. The regularization takes place after the resampling operation and the correction step described in Section 3.

Each box particle \([x_k]\) is characterized by a vector of \(\mathbb{R}^{2d}\) which consists of their center \(c_k\) and their diameter \(\delta_k\). The vector formulation of a box particle is:

\[
\xi_k^T = \left[ c_k^T, \delta_k^T \right] \in \mathbb{R}^{2d}. \quad (44)
\]

The box parameters (center and diameter) can be associated with a random vector \(\xi_k\). Then, a new expression of the density’s approximation can be written, using some kernels \(K\) centered on each box particle \(\xi_k\):

\[
\hat{p}(\xi_k|\mathbf{M}_k) \approx \sum_{i=1}^{N} w_i^k K_h(\xi_k - \xi_i), \quad (45)
\]

where

\[
K_h : \mathbb{R}^{2d} \to \mathbb{R}, \quad K_h(\xi) = \frac{1}{\pi^d} K \left( \frac{\xi}{h} \right), \quad (46)
\]

in the rescaled kernel density \(K(\cdot), h \in \mathbb{R}^+\) is the kernel bandwidth. The kernel density is a symmetric probability density function such that:

\[
\int \xi K(\xi)d\xi = 0, \quad \int \|\xi\|^2 K(\xi)d\xi < \infty. \quad (47)
\]

The kernel \(K(\cdot)\) and bandwidth \(h\) are chosen to minimize the Mean Integrated Square Error (MISE) between the hypothetical posterior density and the corresponding regularized filter’s representation, defined as:

\[
MISE(\hat{p}) = E \left[ (\hat{p}(\xi_k|\mathbf{M}_k) - p(\xi_k|\mathbf{M}_k))^2 d\xi_k \right], \quad (48)
\]

where \(\hat{p}(\xi_k|\mathbf{M}_k)\) denotes the approximation to \(p(\xi_k|\mathbf{M}_k)\) given by \(45\). If all the box particles have the same weight, during the resampling step, an optimal choice of kernel is the bounded Epanechnikov kernel [23],

\[
K_{opt}(\xi) = \begin{cases} \frac{3^{\frac{d+2}{2}}}{2^{d+1}} \left( 1 - \|\xi\|^2 \right) & \text{if } \|\xi\| < 1 \\ 0 & \text{otherwise}, \end{cases} \quad (49)
\]

where \(c_{2d}\) is the volume of the unit hypersphere in \(\mathbb{R}^{2d}\).

The kernel support length is expressed as:

\[
\begin{align*}
\hat{h}_{opt} &= \mu A(K) N^{\frac{1}{2d+1}} \\
A(K) &= \left[ 8c_{2d}^2 (2d + 4)(2\sqrt{\pi})^{2d+1} \right]^{\frac{1}{2d+1}},
\end{align*} \quad (50)
\]

where \(\mu \in [0,1]\) is a tuning parameter, introduced to avoid over-smoothing the density, which would produce divergence.

After kernel smoothing regularization, the box particle kernels mixture is guaranteed to better fit the optimal posterior density. In practice, this will result in an improved estimation accuracy.

### 4.3 BRPF algorithm

The above developments are summarized in Algorithm 1.

**Algorithm 1 Box Regularized Particle Filter**

1. Generate \(N\) box particles \([x_k]\) of empty intersection, associated to weights \(w_0^k = 1/N\).
2. **for** each time-step \(k\) **do**
4. Contract box particles using (15) (if a measurement is available).
5. Update weights using (21).
7. Estimate state \(\mathbf{x}_k\) (23) and its confidence \(\mathbf{P}_k\) (24).
8. **if** (22) is satisfied **then**
9. Use a resampling method (e.g., multinomial resampling) to determine the number of new boxes \(n^t\) per existing box particle.
10. Chose one subdivision dimension \(d^t\) per box particle using (29), (26), and (27).
11. Subdivide each box in \(n^t\) new boxes along its edge \(d^t\).
12. Reset all weights to \(w_0^t = 1/N\).
13. Regularize the box particle cloud by noising each box parameter (44) using the optimal smoothing kernel defined by (49) and (50).
14. **end if**
15. **end for**

### 5 Numerical results, application to terrain aided navigation

To illustrate the behaviour of the resulting filter with non-injective and uncertain measurements, an application to Terrain Aided Navigation (TAN) is presented. The TAN problem is a non-linear and ambiguous estimation problem. To begin, the observation equation involves a non-analytic Digital Elevation Model (DEM)
map, which cannot be linearized in the general case. As a result, Kalman filters (EKF, UKF) are not suitable. Then, the DEM often contains several similar patterns (e.g., peaks and valleys), which makes the observation model non-injective with respect to the state. In addition, the measurement is a scalar value, which does not provide much information about the state. In practice, conventional particle filters often fail to solve this problem [16].

The state vector represents an aerial vehicle coordinate in a geographical frame at time-step \( k \), it consists of:

\[
x_k^T = [p_k^T, v_k^T] \in \mathbb{R}^6,
\]

where \( p_k = [x_k, y_k, z_k]^T \) is the position vector and \( v_k \in \mathbb{R}^3 \) is the velocity vector.

The vehicle dynamical model is assumed to be linear:

\[
x_k = F x_{k-1} + w_k,
\]

where \( F = \begin{bmatrix} I_3 & dt I_3 \\ 0_3 & I_1 \end{bmatrix} \) with \( 0_n \in \mathbb{R}^{n \times n} \) the null matrix and \( dt \) a time-step value. A radar altimeter provides elevation measurements (the relative height \( m_k \), see Fig. 1) along the vehicle trajectory at discrete time values. By comparing these elevations with the DEM, it is possible to reconstruct the absolute position of the aircraft. The DEM gives the absolute elevation as a function of the geographical coordinates \( (x_k, y_k) \). The measurement equation is, as illustrated in Figure 1:

\[
m_k = z_k - \text{DEM}(x_k, y_k) + v_k \in \mathbb{R},
\]

where \( \text{DEM} : \mathbb{R}^2 \rightarrow \mathbb{R} \) is the embedded terrain map and \( v_k \in \mathbb{R} \) is the measurement noise. There is no analytic description of \( \text{DEM} \), which is assumed to be obtained from an embedded terrain map. In this paper, for reproducibility purpose only, the terrain map is analytically generated by the following equation, corresponding to the MATLAB® peaks() function plus a Fourier series:

\[
\text{DEM}(x, y) : \mathbb{R}^2 \mapsto \mathbb{R}
\]

\[
z = \text{peaks}(qx, qy) + \sum_{i=1}^{6} a_i \sin(\omega_i qx) \cos(\omega_i qy),
\]

with \( a_i = \{300, 80, 60, 40, 20, 10\} \), \( \omega_i = \{5, 10, 20, 30, 80, 150\} \), \( \omega_i \in \{4, 10, 20, 40, 90, 150\} \), \( q = 3/(2.96 \times 10^4) \) a scale factor, and \( \text{peaks}(x, y) = 200(3(1 - x)^2 e^{-x^2 - (y+1)^2} - 10(x/5 - x^3 - y^3) e^{-x^2 - y^2} - (1/3) e^{-(x+1)^2 + y^2}) \).

5.1 Simulations overview

In this section, the proposed BRPF algorithm is compared to SIR-PF [9], Monte Carlo Markov Chains

<table>
<thead>
<tr>
<th>Simulation configuration</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial position</td>
<td>([-3.0, -19.2, 1.1] \times 10^4 ) m</td>
</tr>
<tr>
<td>Initial velocity</td>
<td>([211.5, 215.3, 0] ) m/s</td>
</tr>
<tr>
<td>Initial estimate error</td>
<td>([1.0, 1.0, 0.1] \times 10^4 ) m</td>
</tr>
<tr>
<td>Actual process noise</td>
<td>([0.1, 0.1, 0.3] ) m</td>
</tr>
<tr>
<td>Time-step dt</td>
<td>100 ms</td>
</tr>
<tr>
<td>Final time T</td>
<td>100 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurements</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radar-altimeter error (support)</td>
<td>([-45, +45] ) m</td>
</tr>
<tr>
<td>Radar-altimeter update rate</td>
<td>( \Delta_{\text{R-A}} = 100 ) ms</td>
</tr>
<tr>
<td>BPF and BRPF</td>
<td></td>
</tr>
<tr>
<td>Resampling coefficient ( \theta_{\text{eff}} )</td>
<td>0.7</td>
</tr>
<tr>
<td>Normalization parameter ( \lambda )</td>
<td>0.5</td>
</tr>
<tr>
<td>Process noise</td>
<td>None</td>
</tr>
<tr>
<td>SIR-PF</td>
<td></td>
</tr>
<tr>
<td>Resampling coefficient ( \theta_{\text{eff}} )</td>
<td>0.5</td>
</tr>
<tr>
<td>Process noise in position (st.d.)</td>
<td>([10, 10] ) m</td>
</tr>
<tr>
<td>Process noise in velocity (st.d.)</td>
<td>([1, 1] \times 10^{-1} ) m/s</td>
</tr>
<tr>
<td>MCMC</td>
<td></td>
</tr>
<tr>
<td>Burn-in samples</td>
<td>5N</td>
</tr>
<tr>
<td>Process noise in position (st.d.)</td>
<td>([10, 10] ) m</td>
</tr>
<tr>
<td>Process noise in velocity (st.d.)</td>
<td>([1, 1, 10] \times 10^{-1} ) m/s</td>
</tr>
</tbody>
</table>

(MCMC [3]) and BPF [8]. The individual effects of the two improvements presented in section 4 are numerically illustrated, namely the box particles subdivision for resampling (4.1) and the kernel regularization subdivision (4.2).

Table 2 describes the simulation parameters. The number of (box) particles for each filter was chosen to get a similar computation load. The computation time is based on a 1.6 GHz CPU running MATLAB®. A hundred Monte Carlo simulations are run. The first evaluation criterion is the Root Mean Square Error (RMSE) defined by \( \text{RMSE}_X(k) = \sqrt{\frac{1}{N_{MC}} \sum_{j=1}^{N_{MC}} \| \hat{X}_{k,j} - X_{k,j} \|^2} \), where \( N_{MC} = 100 \) is the number of runs. Vector \( X_{k,j} \) stands for actual vector to be estimated. The second evaluation criterion is the mean estimate uncertainty, defined by \( \sigma_X(k) = \frac{1}{N_{MC}} \sum_{j=1}^{N_{MC}} \| \hat{\sigma}_X^j(k) \| \in \mathbb{R}^2 \), where \( \hat{\sigma}_X^j(k) \) contains the diagonal coefficients of the square root of \( \hat{P}_X(k) \).

Table 2 Simulation parameters
Table 3
Simulation results (final time-step)

<table>
<thead>
<tr>
<th></th>
<th>SIR-PF</th>
<th>MCMC</th>
<th>BPF</th>
<th>BRPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>$5 \times 10^4$</td>
<td>$5 \times 10^3$</td>
<td>$10^3$</td>
<td>$10^3$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>$RMSE_x$ (m)</td>
<td>404</td>
<td>292</td>
<td>171</td>
<td>121</td>
</tr>
<tr>
<td>$RMSE_v$ (m/s)</td>
<td>3.7</td>
<td>5.5</td>
<td>3.2</td>
<td>2.8</td>
</tr>
<tr>
<td>$\sigma_x$ (m)</td>
<td>71</td>
<td>187</td>
<td>482</td>
<td>217</td>
</tr>
<tr>
<td>$\sigma_v$ (m/s)</td>
<td>4.1</td>
<td>8.6</td>
<td>4.6</td>
<td>4.0</td>
</tr>
<tr>
<td>Div (%)</td>
<td>4</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Time (ms)</td>
<td>123</td>
<td>356</td>
<td>11</td>
<td>33</td>
</tr>
</tbody>
</table>

5.2 Numerical results

Fig 2 shows the individual effects of each improvement (variance-based box particle subdivision in green and kernel smoothing regularization in red) on the RMSE criterion in comparison to the original BPF (black curve). The variance-based box particle resampling presented in 4.1 allows the filter to restrict its uncertainty and results in a lower error (green curve). The kernel smoothing regularization presented in 4.2 helps to converge more accurately. The combination of these modifications results in a lower estimation error (red curve). Figure 3 illustrates the filter’s behavior for one run (BRPF, $\mu = 0.3$).

Table 3 presents the performance obtained with SIR-PF, MCMC, BPF, and BRPF for several values of $\mu$. For a similar RMSE order of magnitude, the SIR-PF requires 50 times more particles than the BPF, which yields a significantly higher computation time than the BPF and the BRPF. The MCMC method appears more accurate than the SIR-PF with less samples but requires a higher computational load due to the burn-in period (see [3]). SIR-PF and MCMC require a greater computation time per time-step than the desired 100 ms time-step imposed by the measurement rate (10 Hz). With a computation time of only 11 ms and 33 ms, the BPF and the BRPF yield a significantly lower RMSE. BPF and BRPF also produce a lower divergence rate ($\leq 1\%$) than the other methods. This illustrates their ability to robustly tackle real-time applications with non-linear measurements. However, the original BPF yields a conservative final covariance ($\sigma_x = 482$ m and $\sigma_v = 4.8$ m/s). As expected, the results obtained using the covariance-based subdivision resampling strategy (BRPF with $\mu = 0$) are more accurate and less conservative than those obtained using the random subdivision strategy (BPF), resulting in a lower RMSE (121 m versus 171 m) and a lower estimation confidence (217 m versus 482 m). The use of the Kernel Smoothing strategy $\mu > 0$ makes it possible to enhance the estimate accuracy (for $\mu = 0.1$ and $\mu = 0.3$ the RMSE falls from 121 m to 99 m in position and from 2.8 m/s to 2.2 m/s in velocity). As a result, BRPF appears to be able to tackle ambiguous measurements (non-injective observation model) in a more accurate way than previous approaches while meeting real-time requirements.

6 Conclusion

In this work, a state estimation algorithm named Box Regularized Particle Filter (BRPF) is introduced. The algorithm description is more general than the ones presented in previous box particle filters approaches, in terms of dynamical model, observation model and involved uncertainties (Section 3). Two main improvements are introduced in the BPF classic formulation.
(Section 4). The first one provides a general way of subdividing the box particles during the box resampling step (Section 4.1). This formulation is analytic and deterministic. It can be applied to any dynamical and observation models. It leads the box particles to take the shape of the smallest expected variance. As a result, the estimation accuracy is significantly improved compared to the original BPF (see Section 5). The second one is a stochastic posterior kernel smoothing method (Section 4.2). It allows the estimated state density to fit the theoretical state density in an optimal way in the sense of the Mean Integrated Square Error criterion. The performance enhancement resulting from both contributions is demonstrated by numerical simulations on a severely ambiguous and non-linear problem, the Terrain Aided Navigation (see Section 5). Higher accuracy is obtained by the BRPF compared to the BPF in terms of Root Mean Square Error. It is also far more robust than the conventional SIR-PF and the MCMC approach in terms of divergence rate. As a result, BRPF achieves an efficient trade-off between robustness and accuracy.

References