An Efficient Particle Filtering Algorithm based on Ensemble Kalman Filter Proposal Density

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

21-06-2021 / 22-06-2021

CITATION

Tamboli, Prakash (2021): An Efficient Particle Filtering Algorithm based on Ensemble Kalman Filter Proposal Density. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.14813586.v1>

DOI

[10.36227/techrxiv.14813586.v1](https://doi.org/10.36227/techrxiv.14813586.v1)

An Efficient Particle Filtering Algorithm based on Ensemble Kalman Filter Proposal Density

Prakash K. Tamboli, *Member IEEE*, Siddhartha P. Duttagupta, IITB, Kallol Roy, IITM

Abstract—This paper describes a novel Particle Filter algorithm that is both computationally efficient and accurate. The proposed Particle Filter addresses the issue of degeneracy by using the Ensemble Kalman Filter as the proposal density function. The suggested methodology combines the strength of the Ensemble Kalman Filter's Linear Gaussian update with the Monte Carlo technique. The technique considerably decreases processing effort while also resolving the problem or degeneracy inherent in traditional Sequential Importance Algorithms. The problem of Bayesian state estimation in a non-linear/non-Gaussian framework is considered. The proposed algorithm is studied in comparison with the other available proposal densities based on Gaussian approximation. It is demonstrated that the suggested method does not require explicit computation of error covariance for each sample, significantly reducing computational effort. The proposed algorithm's suitability and effectiveness have been thoroughly validated in the simulation with benchmark non-linear estimation problem for various scenarios.

Index Terms—Particle Filter, Sequential Monte Carlo, Ensemble Kalman Filter

I. INTRODUCTION

BAYESIAN estimation is a method to combine the evidence or likelihood with prior knowledge to get an update or posterior probability using Bayes' theorem. Once the posterior probability is available, the "measurement of the central tendency" is used in most cases to estimate the state (variable or parameter). Bayesian state estimation is an important technique in Bayesian statistics and has found applications in areas ranging from science, engineering, medicine, robotics and so on.

In almost all the complex applications of the real world, posterior probability density is never available in closed form, therefore approximate techniques are the only alternatives. Some algorithms are approximations of the process model while some attempt to approximate the state in probability space. In complex applications like meteorology or machine learning, the state and measurement dimensions are enormous, and state evolution model are highly non-linear. Thus, non-linearity, non-Gaussianity and high dimensionality has increasingly become an inescapable elements to take into account in the state estimation to precisely model the dynamics of the physical systems. The state estimation for such complex

scenarios entirely depends upon the accuracy with which the model can be approximated. The solution is straightforward for a simplified case where the system evolves linearly and the posterior density is exactly represented by mean and variance (for a Gaussian posterior). An exact analytical solution can be found for a sequentially evolving linear Gaussian dynamical system, which is known as the *Kalman Filter*.

Over the last few decades many different schemes were developed to solve the aforementioned issues. One of the most popular algorithm is *Extended Kalman Filter* (EKF) which is based on the Taylor series expansion of state evolution function. Other solutions are Gaussian Sum approximation and Grid based filters. However these schemes were still not sufficient to solve complex nonlinear problems. Recent development of high-speed digital computing has opened a new arena of the class of filters known as particle filters or sequential Monte Carlo methods using approximation in probability space [1], [2], [3], [4]. These algorithms have the Bayes' theorem at its heart, which has proven to be the most natural and logical way to combine prior knowledge to make a correct estimate of the posterior. Particle Filtering algorithms represent posterior density by a set of samples. They are drawn from the posterior density function and are represented in combination with the associated weights. However, it may not be always straightforward to draw from the posterior density, hence a proposal density is used to draw the samples. The sample weights are updated in a recursive manner according to the most recently obtained observation data.

Despite being a powerful tool, Particle Filtering still has some serious issues. When proposal density is not properly selected, after a few iterations, all but one sample carries negligible weights. This happens due to lack of overlap between the likelihood and proposal density from which the samples are drawn. The phenomenon in particle filter estimation is also known as "degeneracy". Conventional particle filtering algorithm, also known as "Bootstrap" filter, "Condensation", uses transition prior as the proposal density [1]. This technique is severely effected by degeneracy, since the samples are drawn without the considerations of observation, and there is a very high probability that the many of the sample may fall on the low likelihood are zone. Our goal in this article is to focus on the issue of degeneracy in Particle Filtering and compare different type of proposal densities.

This study provides, first, an overview of the various methods of choosing the importance density functions to draw the samples, Second, we suggest the Ensemble Kalman Filter (EnKF) as a suitable proposal density function. The Ensemble Kalman Filter (EnKF) first appeared in Evensen

Prakash K. Tamboli is Ph.D. from Indian Institute of technology Bombay, Department of Electrical Engineering and with Nuclear Power Corporation of India Limited, e-mail: prakash.tamboli.c2017@iitbombay.org, <https://orcid.org/0000-0003-4034-146X>

Siddhartha. P. Duttagupta is with the Department of Electrical Engineering in Indian Institute of Technology Bombay, Mumbai, India

Kallol Roy is Research Advisor in Indian Institute of Madras, India and former CMD, BHAVINI, India

(1994b) [5] as a new data assimilation technique over the computationally demanding Extended Kalman Filter (EKF). It also appeared in many other articles [6], [7], [8], [9]. The EnKF is an approximate computational technique which represents the probability distribution of system state with the help of a set of samples (ensemble) drawn from the posterior distribution. The samples drawn are restricted by the assumption of Gaussian posterior distribution parametrized by “sample mean” and “sample variance”. The EnKF has become quite popular in Geophysics and is very convenient to implement for real-time applications. In this paper, we present the proposed algorithm’s formulation as well as a comparison of performance with various proposal densities. In terms of sample size and critical characteristics such as convergence, estimation error, and computation time, we draw some useful conclusions for EnKF-based PF.

The rest of the paper is organised as follows. Section II discusses Particle Filtering, as well as different implementation strategies and non-linear state-space estimation difficulties. Section III covers over Ensemble Kalman Filter formulation. Section IV provided the description of the proposed EnKF based Particle Filter methodology. In Section V, we compare the performance of the proposed methodology to that of current approaches in terms of error, convergence, and computing time, and in Section VI we conclude.

II. PARTICLE FILTER

Particle Filters are a class of recursive filter that approximates the posterior density based on Monte Carlo integration [10]. The main idea behind Particle Filtering is to recursively estimate the approximate posterior density in a non-linear environment by using a set of samples and their associated importance weights. Once a new measurement is available, each iteration entails updating the samples and their importance weights. Consider a discrete-time non-linear state evolution model,

$$\mathbf{x}_t = \mathcal{F}(\mathbf{x}_{t-1}, \boldsymbol{\eta}_{t-1}), \quad (1)$$

and measurements as

$$\mathbf{y}_t = \mathcal{G}(\mathbf{x}_t, \boldsymbol{\epsilon}_t), \quad (2)$$

where $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the state transition function. $\mathcal{G} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ is the state observation function. $\mathbf{x}_t, \boldsymbol{\eta}_t \in \mathbb{R}^{n \times 1}$ are n -dimensional system state and process disturbance vectors respectively at a discrete time instant t . Similarly $\mathbf{y}_t, \boldsymbol{\epsilon}_t \in \mathbb{R}^{p \times 1}$ are p -dimensional measurement and noise sequence respectively. It is assumed that $\boldsymbol{\eta}_t$ and $\boldsymbol{\epsilon}_t$ are zero mean white noise sequences, mutually and serially independent to each other with known PDFs.

For simplicity, we assume that the state evolution is a 1st order Markovian process. Assuming we have the posterior density $p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1})$ at the time instant $t-1$ where $\mathbf{Y}_{t-1} = \{\mathbf{y}_j : j = 1, 2, \dots, t-1\}$ represent a set of measurement up to time instant $t-1$. We are also generally aware of the model for evaluating likelihood density function $p(\mathbf{y}_t|\mathbf{x}_t)$. The

prior density $p(\mathbf{x}_t|\mathbf{Y}_{t-1})$ at time t can be evaluated using the following integral

$$p(\mathbf{x}_t|\mathbf{Y}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1})d\mathbf{x}_{t-1} \quad (3)$$

where the probabilistic model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ for the state transition equation using Eq. 1 is described by

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \int \delta(\mathbf{x}_t - \mathcal{F}(\mathbf{x}_{t-1}, \boldsymbol{\eta}_{t-1}))p(\boldsymbol{\eta}_{t-1})d\boldsymbol{\eta}_{t-1} \quad (4)$$

where $\delta(\cdot)$ is the Dirac-delta function. Once the latest measurement $\mathbf{y}_t$ is available, the posterior is evaluated by the Bayesian update

$$p(\mathbf{x}_t|\mathbf{Y}_t) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_{t-1})}{p(\mathbf{y}_t|\mathbf{Y}_{t-1})}, \quad (5)$$

where denominator is normalising constant

$$p(\mathbf{y}_t|\mathbf{Y}_t) = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{Y}_{t-1})d\mathbf{x}_t \quad (6)$$

Following recursion can be built for the sequential case

$$p(\mathbf{x}_t|\mathbf{Y}_t) = p(\mathbf{x}_{t-1}|\mathbf{Y}_{t-1})\frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{Y}_{t-1})}, \quad (7)$$

Further, it is assumed that observation $\mathbf{y}_t$ at any time is conditionally independent of its past values. The posterior density becomes

$$p(\mathbf{x}_t|\mathbf{Y}_t) = p(\mathbf{x}_t|\mathbf{y}_t). \quad (8)$$

This leads to a simplified recursive expression using only the last observation

$$p(\mathbf{x}_t|\mathbf{y}_t) = p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1})\frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{t-1})}, \quad (9)$$

The particle filter approximates the posterior density function using a set of finite samples $\{\mathbf{x}_t^{(i)}; i = 1, 2, \dots, N\}$ with their discrete probability masses $\omega_t^{(i)}$ called *importance weights*, such that

$$p(\mathbf{x}_t|\mathbf{y}_t) \approx \sum_{i=1}^N \omega_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \quad (10)$$

The weights $\omega_t^{(i)}$ are chosen as per the importance density principal. Since it is not always straightforward to draw samples from $p(\mathbf{x}_t|\mathbf{y}_t)$, the sample are drawn from the *importance density*, or *proposal density* $\pi(\mathbf{x}_t|\mathbf{y}_t)$ from which they can be easily drawn. The required conditions for the importance density $\pi(\mathbf{x}_t|\mathbf{y}_t)$ is that it must contain the support of target density $p(\mathbf{x}_t|\mathbf{y}_t)$. The weights can be defined as [11]

$$\omega_t^{(i)} \propto \frac{p(\mathbf{x}_t^{(i)}|\mathbf{y}_t)}{\pi(\mathbf{x}_t^{(i)}|\mathbf{y}_t)} \quad (11)$$

Further factorising of eqn. (11) along with the posterior density recursion established in eqn. (7) lead to a recursive expression for the importance weights [11].

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}, \quad (12)$$

Thus, if we can generate the samples from proposal density $\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)$ and evaluate the weights using (12), we can recursively estimate the approximate posterior density given by (10). The derivation for eqn. (12) is provided in

The next most important thing is to choose the right proposal density. It is an established fact that if there is lack of support points, variance of the weights is bound to increase and a phenomenon known as *degeneracy* occurs. Consequently, all the computation effort is spent to propagate only one sample that decreases the overall estimation accuracy as well as a waste of computational efforts. There are infinite choice of proposal density. An optimal proposal density will keep the variance of sample weights to minimum, and hence all the weights will be almost equal. It is shown that the optimal proposal density is $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$ [12], however, it is not always easy to draw samples from this as stated earlier.

An important step to tackle degeneracy, is re-sampling of the samples according to their importance weights. As the number of significant samples drop considerably, re-sampling is performed to increase the occurrence of samples with more weights. The basic idea is to avoid samples that does not contribute to the estimation accuracy and to utilize computation power more effectively. The pseudo-code for the basic Sequential Importance Re-sampling (SIR) algorithm is described in Algorithm 1.

Algorithm 1: SIR algorithm

```

 $\{\mathbf{x}_t^{(i)}, \omega_t^{(i)}\} = \text{PF}\{\mathbf{x}_{t-1}^{(i)}, \omega_{t-1}^{(i)}, \mathbf{y}_t\}; i = 1, 2, 3, \dots, N$ 
for  $i=1$  to  $N$  do
    Draw  $\mathbf{x}_t$  from  $\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)$ 
     $\tilde{\omega}_t^{(i)} \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}$ 
    Normalise  $\tilde{\omega}_t^{(i)}$ 
    for  $i=1$  to  $N$  do  $\omega_t^{(i)} = \tilde{\omega}_t^{(i)} / \sum_i \tilde{\omega}_t^{(i)}$ 
    Calculate number of effective weight  $N_{eff}$ 
    if  $N_{eff} < N_{Th}$  then
        Re-sample
        for  $i=1$  to  $N$  do
             $\{\mathbf{x}_t^{(i)}, \omega_t^{(i)}\} = \text{RESAMPLE } \{\mathbf{x}_t^{(i)}, \omega_t^{(i)}\}$ 
    Evaluate Sample Mean  $\hat{\mathbf{x}}_t = \sum_i \mathbf{x}_t^{(i)} \omega_t^{(i)}$ 
return

```

A. Bootstrap Particle Filter

An early solution was given by Gordon et. al. (1993) [1], which was to draw the samples from transitional prior $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The technique is known as “bootstrap” or “Condensation” algorithm. With transitional prior as the proposal, the weight recursion simply becomes

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)}).$$
 (13)

The above technique makes computation quite simpler, and we only need to evaluate the likelihood of samples after they are propagate in time through the stochastic process model, to generate new weights.

Although the technique is quite simple to implement, its success heavily depends upon the amount of overlap between the prior and the likelihood. If the prior samples doesn’t fall close to the likelihood area, most of the sample weights will soon have almost no importance, and much of computation will be wasted, causing severe degeneracy.

B. Observation Driven Proposal Densities

To reduce the degeneracy; or in other words, to increase the amount of overlap between the samples and likelihood, it is required to generate the samples from a proposal density that incorporates the most recent measurement. The conventional bootstrap Particle Filter moves blindly without using the latest measurement available to it and hence is prone to severe degeneracy, and sensitive to outliers.

Obviously, the most important point in the design of particle filters is to incorporate the most recent measurement into the proposal. A variant of the SIR algorithm was introduced by Pitt and Shephard (1994) [13] known as Auxiliary Particle Filter (APF) that attempts to draw the samples from a proposal density $p(\mathbf{x}_t|\mathbf{y}_t)$. The APF algorithm is derived with the help of an auxiliary variable - hence its name. The importance density considered in APF is $\pi(\mathbf{x}_t, j|\mathbf{y}_t)$ from which the sample pair $(\mathbf{x}_t^{(i)}, j^{(i)})$ are drawn, where the auxiliary variable j is used as an index. The technique can be considered as re-sampling from one step previous time instant $t - 1$, taking the latest measurement $\mathbf{y}_t$ into consideration. The samples are propagated to evaluate the mean $\mu_t^{(i)}$ for each sample $\mathbf{x}_{t-1}^{(i)}$ at time instant $t - 1$. Here $\mu_k^{(i)}$ is some value that is able to characterize the prior density, although other indicators are also possible. The likelihood $p(\mathbf{y}_t|\mu_t^{(i)})$ is evaluated to calculate the weights that are used for re-sampling. The re-sampled particles are then processed using the generic SIR algorithm, and the weights are updated. The weights produced by this method should be much more evenly distributed, and the estimated state should be close to the true state. However, APF has the drawback of performing well only when the process noise is low. Because a single point cannot characterise the posterior density $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ in the presence of substantial process noise, the APF will yield poor results and be prone to degeneracy due to the poor approximation.

Another class of likelihood-based particle filters draw samples directly from proposal densities based on Kalman filter-based estimators, which are specifically intended for estimating the state in a non-linear environment. Two major variants of Kalman Filter that deal with the non-linear process and measurement models are Extended Kalman Filter (EKF) and Unscented Kalman Filter. There are similar particle filter hybridisation schemes has been in the literature for various applications [14], [15]. Particle Filters variants using these approaches are known as EKF Particle Filter (EPF) and UKF Particle Filter (UPF) [16] respectively. Since these filters recursively update the state using likelihood and provide an estimate of the state, it is quite simple to incorporate them in the SIR algorithm and draw samples. Although they are Gaussian approximation to the posterior $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$, they tend to successfully reduce variance among the importance

weights, and keep the number of effective samples large enough.

The EKF is most widely used estimator for non-linear systems and is the result of a linearisation of the state evolution model using the Taylor series expansion. However the major drawback of EKF is that it can handle only mild non-linearity (approximately linear within a time update) and hence not reliable. Also the algorithm requires computation of the Jacobian which is can be quite computationally intensive for large dimensional systems.

So far, the UKF has proven to be the best version of Kalman Filter for the non-linear process model. It is a deterministic sampling method whereby sampling points are selected strategically. The sampling points when they propagate they capture the first and second moment up to the 3rd order for a Taylor serial expansion of the non-linear function. It is based on the simple concept that it is better to approximate the posterior and use exact function rather than approximating the function and using the exact posterior. Hence incorporation of UKF in SIR algorithm improves the performance greatly.

One of the most important concerns and the main motivation behind our study is the computational effort required in a non-linear state estimation. It can be seen that if we choose to draw samples by propagating through EKF or UKF based estimator, we have to run the estimation algorithm for each sample separately. Although the estimation accuracy will improve significantly but so the computation time. In this paper we propose to integrate the Ensemble Kalman Filter (EnKF) into the Particle filter due to their similarity and also due to numerous advantages of EnKF. The EnKF similar to the particle Filter algorithm works on a set of sample called as ensemble, and is based on the Monte-Carlo integration. The subsequent sections are dedicated to the design of an EnKF based Particle Filter algorithm that greatly reduces the computation without compromising the accuracy.

III. ENSEMBLE KALMAN FILTER

The EnKF is very similar to Particle Filter in a way that a continuous variable (states) defined by posterior density function $p(\mathbf{x}_t|\mathbf{y}_t)$ is represented using discrete random samples except the associated importance weights. The basic distinction between EnKF and Particle Filter is that with EnKF, the samples are updated by moving them toward the likelihood area once the current measurement is received. The Particle Filter, on the other hand, recalculates weights depending on their likelihood density making EnKF's tracking ability better than basic Particle Filtering algorithm since the samples always overlap with likelihood. There are some recent development in Ensemble Kalman Filtering such as Augmented EnKF [17] to overcome the limitations of linear updating. In EnKF, the error statistics are predicted by Monte Carlo integration to solve the Fokker-Plank equation. Consider a discrete-time non-linear state evolution model,

$$\mathbf{x}_t = \mathcal{F}(\mathbf{x}_{t-1}, \boldsymbol{\eta}_{t-1}), \quad (14)$$

and measurements as

$$\mathbf{y}_t = \mathcal{G}(\mathbf{x}_t, \boldsymbol{\epsilon}_t), \quad (15)$$

where $\mathbf{x}_t, \boldsymbol{\eta}_t \in \mathbb{R}^{n \times 1}$ are n-dimensional system state and process disturbance respectively at a discrete time instant t . $\mathbf{y}_t, \boldsymbol{\epsilon}_t \in \mathbb{R}^{p \times 1}$ are p-dimensional measurement variable and noise signal respectively. It is assumed that $\boldsymbol{\eta}_t$ and $\boldsymbol{\epsilon}_t$ are mutually and serially independent sequence of data. The goal is to obtain the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_t)$ at time t given the measurement data up to point t . Once the posterior distribution is available, an updated state estimate $\mathbf{x}_{t-1}$ can be easily obtained.

Assume that the samples $\mathbf{x}_{t-1}^1, \mathbf{x}_{t-1}^2 \dots \mathbf{x}_{t-1}^N$ are drawn from the Gaussian distribution at a discrete time $t-1$ such that $\mathbf{x}_{t-1}^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ where $\boldsymbol{\mu}_{t-1}$ and $\boldsymbol{\Sigma}_{t-1}$ are *sample mean* and *sample variance* at time t . The samples from the forecast distribution can be obtained by simply applying the state evolution as

$$\mathbf{x}_{t|t-1}^{(i)} = \mathcal{F}(\mathbf{x}_{t-1|t-1}^{(i)}, \boldsymbol{\eta}_{t-1}^{(i)}) \quad i = 1, 2, \dots, N \quad (16)$$

The sample mean of the forecast state is obtained as

$$\boldsymbol{\mu}_{t|t-1} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_{t|t-1}^{(i)}. \quad (17)$$

Similarly, the forecast measurement and the corresponding sample mean can be obtained as

$$\mathbf{y}_{t|t-1}^{(i)} = \mathcal{G}(\mathbf{x}_{t|t-1}^{(i)}, \boldsymbol{\epsilon}_t^{(i)}), \quad (18)$$

$$\bar{\mathbf{y}}_{t|t-1} = \frac{1}{N} \sum_{i=1}^N \mathbf{y}_{t|t-1}^{(i)}. \quad (19)$$

Let $\boldsymbol{\varepsilon}_t^{(i)}$ and $\mathbf{e}_t^{(i)}$ represent the state and measurement estimation errors at time t

$$\boldsymbol{\varepsilon}_t^{(i)} = \mathbf{x}_{t|t-1}^{(i)} - \boldsymbol{\mu}_{t|t-1}, \quad (20)$$

$$\mathbf{e}_t^{(i)} = \mathbf{y}_{t|t-1}^{(i)} - \bar{\mathbf{y}}_{t|t-1}. \quad (21)$$

The Kalman gain is evaluated by using the following covariance matrices

$$P_{\varepsilon\varepsilon} = \frac{1}{N-1} \sum_{i=1}^N \{\boldsymbol{\varepsilon}_t^{(i)}\} \{\boldsymbol{\varepsilon}_t^{(i)}\}^T \quad (22)$$

$$P_{ee} = \frac{1}{N-1} \sum_{i=1}^N \{\mathbf{e}_t^{(i)}\} \{\mathbf{e}_t^{(i)}\}^T \quad (23)$$

$$K_t = P_{\varepsilon e} \{P_{ee}\}^{-1} \quad (24)$$

The updated samples can be obtained as

$$\mathbf{x}_{t|t}^{(i)} = \mathbf{x}_{t|t-1}^{(i)} + K_t \left(\mathbf{y}_t - \mathbf{y}_{t|t-1}^{(i)} \right) \quad (25)$$

The samples thus obtained represent the posterior distribution of the state at time t . Notice that the linear update equation (25) is the same as the one used for analytical Kalman Filtering, with the exception that the predicted observation is perturbed by the observation noise. The analysed samples can be said to be distributed as $\mathcal{N}(\mathbf{x}_t^i : \boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t})$, where $\boldsymbol{\Sigma}_{t|t}$ is the empirical variance computed from analysed samples.

IV. ENSEMBLE KALMAN FILTER PARTICLE FILTER

Using the descriptions given in previous sections, we now present a hybrid SIR algorithm that uses EnKF as the proposal density. Because the basic EnKF works with samples with equal weights, it is relatively more straight forward to incorporate into the SIR algorithm if we choose to re-sample at each iteration. The re-sampling after the weight update stage will result in all weights being equal. However, we present here an alternative solution that employs a slightly modified version of EnKF that works with weighted samples and eliminates the need for weight equalisation at each iteration.

We will assume that the evolution of state vector is described as per the model given in Eq. 1 and 2. As described in Section-II, let the posterior density $p(\mathbf{x}_t|\mathbf{y}_t)$ at time instant t be again represented by samples $\mathbf{x}_t^{(i)}$ and the associated weights $\omega_t^{(i)}$. For sampling from EnKF proposal, the samples $\mathbf{x}_t^{(i)}$ additionally serve as ensemble members. The ensemble members are then displaced as per the following non-linear equation:

$$\mathbf{x}_{t|t-1}^{(i)} = \mathcal{F}(\mathbf{x}_{t-1|t-1}^{(i)}, \boldsymbol{\eta}_{t-1}^{(i)}). \quad (26)$$

The predicted ensemble measurement $\mathbf{y}_t^{(i)}$ perturbed with Gaussian noise $\boldsymbol{\epsilon}_t^{(i)} \sim \mathcal{N}(0, \mathbf{R})$ can be obtained by passing the forecasted ensemble members through the measurement equation

$$\mathbf{y}_{t|t-1}^{(i)} = \mathcal{G}(\mathbf{x}_{t|t-1}^{(i)}, \boldsymbol{\epsilon}_t^{(i)}). \quad (27)$$

The forecasted sample mean of the weighted states and measurement can be obtained as

$$\boldsymbol{\mu}_{t|t-1} = \frac{\sum_{i=1}^N \omega_{t-1}^{(i)} \mathbf{x}_{t|t-1}^{(i)}}{\sum_{i=1}^N \omega_{t-1}^{(i)}}, \quad (28)$$

$$\bar{\mathbf{y}}_{t|t-1} = \frac{\sum_{i=1}^N \omega_{t-1}^{(i)} \mathbf{y}_{t|t-1}^{(i)}}{\sum_{i=1}^N \omega_{t-1}^{(i)}}. \quad (29)$$

Let $\varepsilon_t^{(i)}$ and $e_t^{(i)}$ represent the state and measurement estimation errors at time t

$$\varepsilon_t^{(i)} = \mathbf{x}_{t|t-1}^{(i)} - \bar{\mathbf{x}}_{t|t-1}, \quad (30)$$

$$e_t^{(i)} = \mathbf{y}_{t|t-1}^{(i)} - \bar{\mathbf{y}}_{t|t-1}. \quad (31)$$

Following covariance matrices for weighted samples can be evaluated (with Bessel's correction)

$$P_{\varepsilon e_t} = \frac{\sum_{i=1}^N \omega_{t-1}^{(i)}}{\sum_{i=1}^N \omega_{t-1}^{(i)} - \left(\sum_{i=1}^N \omega_{t-1}^{(i)}\right)^2} \sum_{i=1}^N \omega_{t-1}^{(i)} \varepsilon_t^{(i)} \{\mathbf{e}_t^{(i)}\}^T \quad (32)$$

$$P_{ee_t} = \frac{\sum_{i=1}^N \omega_{t-1}^{(i)}}{\sum_{i=1}^N \omega_{t-1}^{(i)} - \left(\sum_{i=1}^N \omega_{t-1}^{(i)}\right)^2} \sum_{i=1}^N \omega_{t-1}^{(i)} \mathbf{e}_t^{(i)} \{\mathbf{e}_t^{(i)}\}^T \quad (33)$$

If the weights $\omega_{t-1}^{(i)}$ are chosen to be normalised such that

$$\sum_{i=1}^N \omega_{t-1}^{(i)} = 1. \quad (34)$$

The covariance matrices can be redefined as

$$P_{\varepsilon e_t} = \frac{1}{1 - \left(\sum_{i=1}^N \omega_{t-1}^{(i)}\right)^2} \sum_{i=1}^N \omega_{t-1}^{(i)} \varepsilon_t^{(i)} \{\mathbf{e}_t^{(i)}\}^T \quad (35)$$

$$P_{ee_t} = \frac{1}{1 - \left(\sum_{i=1}^N \omega_{t-1}^{(i)}\right)^2} \sum_{i=1}^N \omega_{t-1}^{(i)} \mathbf{e}_t^{(i)} \{\mathbf{e}_t^{(i)}\}^T \quad (36)$$

The Kalman gain can be evaluated using the above matrices

$$K_t = P_{\varepsilon e_t} \{P_{ee_t}\}^{-1} \quad (37)$$

The updated state (analysed ensemble members) are obtained by the following linear combination equation

$$\mathbf{x}_{t|t}^{(i)} = \mathbf{x}_{t|t-1}^{(i)} + K_t (\mathbf{y}_t - \mathbf{y}_{t|t-1}^{(i)}) \quad (38)$$

The resultant samples can be said to be distributed as per the proposal distribution

$$p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) = \mathcal{N}(\mathbf{x}_t^{(i)} : \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_{t|t}), \quad (39)$$

where

$$\boldsymbol{\mu}_t = \sum_{i=1}^N \omega_{t-1}^{(i)} \mathbf{x}_{t|t-1}^{(i)}. \quad (40)$$

The weight update equation in Particle Filter can be now written as

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)})}{\mathcal{N}(\mathbf{x}_t^{(i)}; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_{t|t})}, \quad (41)$$

Thus, the proposal density of EnKFPF is obtained using a common empirical mean and variance, as opposed to EPF or UPF, where the mean $\boldsymbol{\mu}_t^{(i)}$ and variance $\boldsymbol{\Sigma}_t^{(i)}$ for each sample are evaluated for each sample $\mathbf{x}_{t|t}^{(i)}$. The proposal density for EPF and UPF is given by

$$p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) = \mathcal{N}(\mathbf{x}_t^{(i)} : \boldsymbol{\mu}_t^{(i)}, \boldsymbol{\Sigma}_t^{(i)}) \quad (42)$$

Another crucial component of particle filtering is the convergence property, which assures that the estimate error approaches the precise analytical answer as the number of samples approaches infinity. The convergence in Particle Filtering has been discussed in general by Crisan and Doucet (2002) [18], [19], [20], and [21]. The sufficient condition for empirical convergence for a Particle Filter in L^p -sense requires to assume that the process and observation models are both bounded, as are the importance weights (unnormalized). The following conditions must be met for any Borel test function $\phi(\mathbf{x})$.

$$E \left[\left| \int \phi(\mathbf{x}) p(\mathbf{x} | \mathbf{y}) d\mathbf{x} - \sum_{i=1}^N \omega_t^{(i)} \phi(\mathbf{x}_t^{(i)}) \right|^p \right] \leq c_k \frac{\|\phi\|^p}{N^{\frac{p}{2}}} \quad (43)$$

for some constant c_k . The suggested EnKFPF method, like the EPF and UPF algorithms, pulls samples that adequately overlap the likelihood $p(\mathbf{y}_t | \mathbf{x}_t^{(i)})$ and hence weights are not expected to grow unbound. The EnKFPF algorithm thus fulfils the sufficient condition defined above.

The pseudo-code for the basic Sequential Importance Re-sampling (SIR) algorithm is described in Algorithm 2.

Algorithm 2: EnKFPPF algorithm

```

 $\{\mathbf{x}_t^{(i)}, \omega_t^{(i)}\} = \text{EnKFPPF}\{\mathbf{x}_{t-1}^{(i)}, \omega_{t-1}^{(i)}, \mathbf{y}_t\};$ 
 $i = 1, 2, 3, \dots, N$ 
for  $i=1$  to  $N$  do
    Draw  $\mathbf{x}_t$  from  $\mathcal{N}(\mathbf{x}_t^{(i)} : \boldsymbol{\mu}_t, \Sigma_t | \mathbf{y}_t)$  as per eqn. (39)
     $\tilde{\omega}_t^{(i)} \propto \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)})}{\mathcal{N}(\mathbf{x}_t^{(i)} : \boldsymbol{\mu}_t, \Sigma_t | \mathbf{y}_t)}$ 
    Normalise  $\tilde{\omega}_t^{(i)}$ 
    for  $i=1$  to  $N$  do  $\omega_t^{(i)} = \tilde{\omega}_t^{(i)} / \sum_i \tilde{\omega}_t^{(i)}$ 
    Calculate number of effective weight  $N_{eff}$ 
    if  $N_{eff} < N_{Th}$  then
        Re-sample
        for  $i=1$  to  $N$  do
             $\{\mathbf{x}_t^{(i)}, \omega_t^{(i)}\} = \text{RESAMPLE}\{\mathbf{x}_t^{(i)}, \omega_t^{(i)}\}$ 
    Evaluate Sample Mean  $\hat{\mathbf{x}}_t = \sum_i \mathbf{x}_t^{(i)} \omega_t^{(i)}$ 
return

```

V. RESULTS AND DISCUSSIONS

In this section, we present numerical results illustrating performance of the proposed EnKFPPF algorithm for various scenarios, and compare with EKF and UKF proposals density based Particle Filtering algorithms. We consider the following non-linear time series model that is often used in the literature to compare numerical filter techniques [22], [1].

$$x_t = \frac{x_{t-1}}{2} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + \eta_t, \quad (44)$$

$$y_t = \frac{x_t^2}{20} + \epsilon_t, \quad (45)$$

where $\eta_t \sim \mathcal{N}(0, \Sigma_\eta)$ and $\epsilon_t \sim \mathcal{N}(0, \Sigma_\epsilon)$ represent zero mean Gaussian process and measurement noise. The above state evolution model quickly bifurcates the prior $p(x_t | x_{t-1})$ transforming it into a bi-model density function if the posterior density $p(x_{t-1} | y_{t-1})$ is assumed to be zero mean Gaussian. It's also interesting to note that when the measurement yields negative values, the likelihood density's shape shifts from bi-model to zero mean uni-model. For such a bi-modal, non-linear scenario, any algorithms based on the assumption of Gaussianity involving linear update may generally result in a less accurate estimate and will quickly diverge.

The first set of simulations were run with Gaussian assumptions for both the process and the observation model for various sample sizes and uncertainties. For simplicity, the simulations were run without the Metro-police Hastings step. The process and measurement error are assumed to be zero mean white Gaussian with $\sigma_\eta = 1$ and $\sigma_\epsilon = 1$. The results for all three algorithms are demonstrated in Fig. 1, for sample sizes = 1000. For this particular case, UPF appears to be providing better estimate than EnKFPPF. For further quantification of the convergence in MSE error, simulations with various sample sizes were carried out. The results are shown in Fig. 2.

Fig. 1: Plot of estimated state vs true state for sample size=1000, $\sigma_\eta = 10$ (Gaussian) and $\sigma_\epsilon = 10$ (Gaussian)

Fig. 2 illustrates the efficacy of proposed EnKFPPF algorithm along with EPF and UPF in terms of error convergence as a function of sample size for process and observation noise standard deviations σ_η and σ_ϵ equal to 1. The larger sample sizes is showing to improve the results as expected and asymptotically converges to certain value as the sample size is infinite ($N \rightarrow \infty$). In case of EnKFPPF, all the ensemble member are shifted towards the observation using the empirical covariances. As a results the EnKFPPF is expected to generate proposal densities $q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_t)$ that sufficiently overlaps with the $p(\mathbf{y}_t | \mathbf{x}_t)p(\mathbf{x}_t | \mathbf{x}_{t-1})$ resulting into faster convergence.

It is reasonable to assume that the MSEs of all the algorithms converge to the analytical method as N approaches to infinity ($N \rightarrow \infty$). It can be observed that the UPF provide better results for almost all case as compare with EnKFPPF. This is due to the fact that UPF propagates each sample through UKF estimator and hence samples are drawn from individually evaluated density parameters. When large number of samples are used, however, it is clear that EnKF is more efficient in achieving practically identical accuracy. When compared to EPF, both EnKFPPF and UPF perform much better. The inaccurate approximation of the extremely non-linear model is to account for the EPF's poor estimation accuracy.

Fig. 2: Root Mean Squared (RMS) Error vs sample size for EPF, UPF and EnKFPP for $\sigma_\eta=10$, $\sigma_\epsilon=10$

The computation time for all the three algorithm and MSE values for 1000 samples with $\sigma_\eta = 1$ and $\sigma_\epsilon = 1$ is shown in terms of a bar graph in Fig. 3. The crucial element to note here is the EnKFPP's substantially shorter computation time, which corresponds to the assertions stated in the preceding sections. As state earlier, the fact that EnKFPP does not run the computationally intensive estimation algorithms for each sample and instead uses empirical variances results in a significant reduction in computation time. It should be noticed that the EPF has a computation time that is nearly identical to the EnKFPP. However, the computation time in EPF is significantly influenced by the model's structure and dimensions, and hence can vary greatly depending on the case.

Fig. 3: Bar graph of Normalised estimation accuracy and computation time for all three algorithms (for 1000 samples and process and noise uncertainty and $\sigma_\eta, \sigma_\epsilon = 1$)

Another set of trials was conducted by changing the process uncertainty from Gaussian to non-Gaussian density while maintaining Gaussian measurement noise. We choose the process noise uncertainty to be Gamma distributed with parameters $\alpha = 3$ and $\theta = 1/2$. The mean and variance are $\alpha\theta = 3/2$ and $\alpha\theta^2 = 3/4$, respectively, which are used to generate estimated states in the EPF and UPF methods. The EnKFPP, on the other hand, makes use of the empirical variances calculated from the samples. With all three algorithms, the estimated states are presented in Figure-4 for a sample size of 1000. The analysis reveal that UPF gives superior accuracy for small measurement noise in case where process uncertainty is described by non-Gaussian non-zero mean density function case. Due to broader overlap with likelihood, the EnKFPP outperforms UPF in terms of accuracy and computation time for large measurement noise values (e.g., $\sigma_\epsilon = 10$). Fig. 5 illustrates the variation of MSE error verses increasing measurement noise for a sample size of 1000.

For a clear understanding, we provided the estimation results for several Gaussian and non-Gaussian scenarios Table-I, which is in terms of MSE error and computation time averaged over 100 trials. Based on the inaccuracy and computation time in the table, EnKFPP looks to be an efficient alternative that provides both accuracy and a significant reduction in computing cost. EnKFPP's accuracy improves as the level of uncertainty increases, which is another desirable aspect.

Fig. 4: Plot of estimated state vs true state for sample size=50, $\sigma_\eta = 3/4$ (Gamma) and $\sigma_\epsilon = 10$ (Gaussian)

TABLE I: Comparison between different proposal densities

Particle Filtering algorithm	Process Noise	Measurement Noise Gaussian	Statistical MSE	Computation time (Seconds)
EPF	Gaussian $\sigma_\eta = 1$	$\sigma_\epsilon = 1$	9.3634	5.3525
UPF	Gaussian $\sigma_\eta = 1$	$\sigma_\epsilon = 1$	4.2977	13.0064
EnKF-PF	Gaussian $\sigma_\eta = 1$	$\sigma_\epsilon = 1$	5.4486	2.6637
EPF	Gaussian $\sigma_\eta = 1$	$\sigma_\epsilon = 10$	11.6119	5.3525
UPF	Gaussian $\sigma_\eta = 1$	$\sigma_\epsilon = 10$	6.8647	13.0064
EnKF-PF	Gaussian $\sigma_\eta = 1$	$\sigma_\epsilon = 10$	10.2963	2.6637
EPF	Gamma $\sigma_\eta = 3/4$	$\sigma_\epsilon = 1$	14.0728	5.3921
UPF	Gamma $\sigma_\eta = 3/4$	$\sigma_\epsilon = 1$	6.8512	13.2031
EnKF-PF	Gamma $\sigma_\eta = 3/4$	$\sigma_\epsilon = 1$	9.1584	2.9731
EPF	Gamma $\sigma_\eta = 3/4$	$\sigma_\epsilon = 10$	16.8333	5.3921
UPF	Gamma $\sigma_\eta = 3/4$	$\sigma_\epsilon = 10$	12.6052	13.2031
EnKF-PF	Gamma $\sigma_\eta = 3/4$	$\sigma_\epsilon = 10$	9.7171	2.9731

Fig. 5: MSE error verses measurement uncertainty for sample size of 1000

VI. CONCLUSION

In this article, a version of Ensemble Kalman Filter-based proposal density has been suggested in order to realise an efficient Particle Filter that does not require large computational effort. The proposed algorithm draws the samples straight away from the ensemble mean and variance evaluated in EnKF's analysis step and uses them to update the weights. The proposed approach thus effectively combines the benefits of both analytical and Monte Carlo approaches to improve accuracy. We ran several scenarios on the benchmark non-linear problem to demonstrate that the proposed methodology is on par with the UPF in terms of accuracy and convergence

while requiring far less computational effort. The deterministic shift in the samples towards measurement makes it suitable for many real world on-line application where tracking ability is required along-with lower computation effort.

Finally, we note that the efficacy of our proposed algorithm is shown for specific problem using scalar-non-linear case only with Gaussian and bi-modal uncertainties and can be extended to multi-modal cases. We believe that the proposed method may be further studies in details with respect to the large dimensional problems in general.

APPENDIX A

DERIVATION FOR THE PARTICLE FILTER WEIGHT RECURSION EQUATION

The sample weights drawn from the proposal density $\pi(\mathbf{x}_t|\mathbf{y}_t)$ instead of the desired posterior density function $p(\mathbf{x}_t|\mathbf{y}_t)$ is defined as [11]

$$\omega_t \propto \frac{p(\mathbf{x}_t|\mathbf{y}_t)}{\pi(\mathbf{x}_t|\mathbf{y}_t)} \quad (46)$$

The numerator in the above equation (46) is given by

$$p(\mathbf{x}_t|\mathbf{y}_t) = \frac{p(\mathbf{y}_t|\mathbf{x}_t|\mathbf{y}_{t-1}) p(\mathbf{x}_t|\mathbf{y}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{t-1})} \quad (47)$$

$$= \frac{p(\mathbf{y}_t|\mathbf{x}_t|\mathbf{y}_{t-1}) p(\mathbf{x}_t|\mathbf{x}_{t-1}|\mathbf{y}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{t-1})} \times p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}) \quad (48)$$

$$= \frac{p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{t-1})} \times p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}) \quad (49)$$

$$\propto p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}). \quad (50)$$

The proposal density is be chosen to be factorised as

$$\pi(\mathbf{x}_t|\mathbf{y}_t) = \pi(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) \pi(\mathbf{x}_{t-1}|\mathbf{y}_{t-1}). \quad (51)$$

By substituting eqn. (50) and eqn (51) in eqn. (7), we get the following equation

$$\omega_t \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1})}{\pi(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) \pi(\mathbf{x}_{t-1}|\mathbf{y}_{t-1})} \quad (52)$$

The above can be re-written in a recursive form for weight updation as below

$$\omega_t \propto \omega_{t-1} \frac{p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1})}{\pi(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1})} \quad (53)$$

REFERENCES

- [1] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE proceedings F (radar and signal processing)*, vol. 140, no. 2. IET, 1993, pp. 107–113. 1, 3, 6
- [2] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, 1999. 1
- [3] A. F. Smith and A. E. Gelfand, "Bayesian statistics without tears: a sampling–resampling perspective," *The American Statistician*, vol. 46, no. 2, pp. 84–88, 1992. 1
- [4] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and bayesian missing data problems," *Journal of the American statistical association*, vol. 89, no. 425, pp. 278–288, 1994. 1
- [5] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, vol. 99, no. C5, pp. 10 143–10 162, 1994. 2

- [6] —, “The ensemble kalman filter: Theoretical formulation and practical implementation,” *Ocean dynamics*, vol. 53, no. 4, pp. 343–367, 2003. 2
- [7] G. Burgers, P. Jan van Leeuwen, and G. Evensen, “Analysis scheme in the ensemble kalman filter,” *Monthly weather review*, vol. 126, no. 6, pp. 1719–1724, 1998. 2
- [8] P. L. Houtekamer and H. L. Mitchell, “Data assimilation using an ensemble kalman filter technique,” *Monthly Weather Review*, vol. 126, no. 3, pp. 796–811, 1998. 2
- [9] —, “A sequential ensemble kalman filter for atmospheric data assimilation,” *Monthly Weather Review*, vol. 129, no. 1, pp. 123–137, 2001. 2
- [10] A. Doucet, N. De Freitas, and N. Gordon, “An introduction to sequential monte carlo methods,” in *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 3–14. 2
- [11] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002. 2, 8
- [12] A. Doucet, S. Godsill, and C. Andrieu, “On sequential monte carlo sampling methods for bayesian filtering,” *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000. 3
- [13] M. K. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filters,” *Journal of the American statistical association*, vol. 94, no. 446, pp. 590–599, 1999. 3
- [14] Z. Jiang, W. Zhou, H. Li, Y. Mo, W. Ni, and Q. Huang, “A new kind of accurate calibration method for robotic kinematic parameters based on the extended kalman and particle filter algorithm,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3337–3345, 2018. 3
- [15] P. K. Tamboli, S. P. Duttgupta, and K. Roy, “Neutron flux estimation using particle-h filter for fast transient tracking,” *Annals of Nuclear Energy*, vol. 90, pp. 115–122, 2016. 3
- [16] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. A. Wan, “The unscented particle filter,” *Adv. Neural Inform. Process. Syst.*, 2000. 3
- [17] T. Sun and M. Xin, “Bearings-only tracking using augmented ensemble kalman filter,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 1009–1016, 2020. 4
- [18] D. Crisan and A. Doucet, “A survey of convergence results on particle filtering methods for practitioners,” *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 736–746, 2002. 5
- [19] A. Bain and D. Crisan, *Fundamentals of stochastic filtering*. Springer Science & Business Media, 2008, vol. 60. 5
- [20] X.-L. Hu, T. B. Schon, and L. Ljung, “A basic convergence result for particle filtering,” *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1337–1348, 2008. 5
- [21] O. Cappé, E. Moulines, and T. Rydén, *Inference in hidden Markov models*. Springer Science & Business Media, 2006. 5
- [22] G. Kitagawa, “Non-gaussian statespace modeling of nonstationary time series,” *Journal of the American statistical association*, vol. 82, no. 400, pp. 1032–1041, 1987. 6