A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking *

Simon Maskell
Qinetiq Ltd and Cambridge University Engineering Department, UK.
s.maskell@signal.qinetiq.com

Neil Gordon
QinetiQ Ltd, UK.
n.gordon@signal.qinetiq.com

September 2001

1 Introduction

Bayesian methods provide a rigorous general framework for dynamic state estimation problems. The Bayesian approach is to construct the pdf of the state vector based on all available information. This pdf summarises the current state of knowledge about the state vector and from it the optimal (with respect to whatever cost function the user chooses) course of action can be determined. For the linear/Gaussian estimation problem, the required pdf remains Gaussian at every iteration of the filter, and the Kalman filter relations propagate and update the mean and covariance of the distribution. For a nonlinear/ non-Gaussian problem there is in general no analytic (closed form) expression for the required pdf. However, for many applications these modelling assumptions are strongly implied by considerations of realism. For example, bearings only tracking in a Cartesian co-ordinate system implies a nonlinear relationship between the state and the measurement. Similarly, if measurement sensors produce occasional gross errors (eg radar glint), a heavy-tailed non-Gaussian error model could be appropriate.

In the following notes we begin with a description of the nonlinear/non-Gaussian tracking problem and its optimal Bayesian solution. Since the optimal solution is intractable, several different approximation strategies are then described. These approaches include the extended Kalman filter and particle filters. These notes are of a tutorial nature and so, to facilitate easy implementation, 'pseudo-code' for algorithms has been included at relevant points.

2 Nonlinear Bayesian Tracking

To define the problem of tracking, consider the evolution of the state sequence $\{\mathbf{x}_k, k \in \mathbb{N}\}$ of a target, given by

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \tag{1}$$

where $\mathbf{f}_k : \Re^{n_x} \times \Re^{n_v} \to \Re^{n_x}$ is a possibly nonlinear function of the state $\mathbf{x}_{k-1}$, $\{\mathbf{v}_{k-1}, k \in \mathbb{N}\}$ is an i.i.d process noise sequence, n_x, n_v are dimensions of the state and process noise vectors, respectively and $\mathbb{N}$ is the set of natural numbers. The objective of tracking is to recursively estimate $\mathbf{x}_k$ from measurements

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k), \tag{2}$$

 This paper is a condensed version of [1].

where $\mathbf{h}_k : \Re^{n_x} \times \Re^{n_n} \to \Re^{n_z}$ is a possibly nonlinear function, $\{\mathbf{n}_k, k \in \mathbb{N}\}$ is an i.i.d measurement noise sequence, and n_z, n_n are dimensions of the measurement and measurement noise vectors, respectively. In particular, we seek filtered estimates of $\mathbf{x}_k$ based on the set of all available measurements $\mathbf{z}_{1:k} = \{\mathbf{z}_i, i = 1, \ldots, k\}$ up to time k.

From a Bayesian perspective, the tracking problem is to recursively calculate some degree of belief in the state $\mathbf{x}_k$ at time k, taking different values, given the data $\mathbf{z}_{1:k}$. Thus, it is required to construct the pdf $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. It is assumed that the initial pdf, $p(\mathbf{x}_0|\mathbf{z}_0) \equiv p(\mathbf{x}_0)$, of the state vector, also known as the prior, is available ($\mathbf{z}_0$ being the set of no measurements). Then, in principle, the pdf $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ may be obtained recursively in two stages: prediction and update.

Suppose that the required pdf $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ at time $k-1$ is available. The prediction stage involves using the system model (1) to obtain the prior pdf of the state at time k via the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \tag{3}$$

At time step k, a measurement $\mathbf{z}_k$ becomes available, and this may be used to update the prior (update stage) via Bayes' rule:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}, \tag{4}$$

where the normalising constant

$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) d\mathbf{x}_k \tag{5}$$

depends on the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$. In the update stage (4), the measurement $\mathbf{z}_k$ is used to modify the prior density to obtain the required posterior density of the current state.

The recurrence relations (3) and (4) form the basis for the optimal Bayesian solution. This recursive propagation of the posterior density is only a conceptual solution in that in general, it cannot be determined analytically. Solutions do exist in a restrictive set of cases, including the Kalman filter. We now describe how, when the analytic solution is intractable, extended Kalman filters and particle filters approximate the optimal Bayesian solution.

3 Extended Kalman Filter

The most well known, and certainly the most popular, approach to implementing recursive nonlinear filters is the extended Kalman filter (EKF), see [19]. This approach is applicable to nonlinear models with additive Gaussian noise. It is a linearisation technique based on a first order Taylor series expansion of the nonlinear system and measurement functions about the current estimate of the state. This requires that both $\mathbf{f}_k$ and $\mathbf{h}_k$ be differentiable functions of their vector arguments.

$$\begin{aligned}
p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) &\approx \mathcal{N}(\mathbf{x}_{k-1}; m_{k-1|k-1}, P_{k-1|k-1}) && (6)\\
p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) &\approx \mathcal{N}(\mathbf{x}_k; m_{k|k-1}, P_{k|k-1}) && (7)\\
&= \mathcal{N}(\mathbf{x}_k; \mathbf{f}_k(m_{k-1|k-1}), Q_{k-1} + \hat{F}_k P_{k-1|k-1} \hat{F}_k^T) && (8)\\
p(\mathbf{x}_k|\mathbf{z}_{1:k}) &\approx \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) \\
&= \mathcal{N}(\mathbf{x}_k; m_{k|k-1} + K_k(\mathbf{z}_k - \mathbf{h}_k(m_{k|k-1})), P_{k|k-1} - K_k \hat{H}_k P_{k|k-1}) && (9)
\end{aligned}$$

where $\mathbf{f}_k(.)$ and $\mathbf{h}_k(.)$ are nonlinear functions and $\hat{F}_k$ and $\hat{H}_k$ are local linearisations of these nonlinear functions (ie. matrices):

$$\hat{F}_k = \frac{d\mathbf{f}_k(x)}{dx}\Big|_{x=m_{k-1|k-1}} \tag{10}$$

$$\hat{H}_k = \frac{d\mathbf{h}_k(x)}{dx}|_{x=m_{k|k-1}} \quad (11)$$

$$S_k = \hat{H}_k P_{k|k-1} \hat{H}_k^T + R_k \quad (12)$$

$$K_k = P_{k|k-1} \hat{H}_k^T S_k^{-1} \quad (13)$$

The EKF as described above utilises the first term in a Taylor expansion of the nonlinear function. A higher order EKF that retains further terms in the Taylor expansion exists, but the additional complexity has prohibited its widespread use.

Recently, the unscented transform has been used in an EKF framework [20, 31, 32]. The resulting filter, known as "Unscented Kalman Filter", considers a set of points that are deterministically selected from the Gaussian approximation to $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. These points are all propagated through the true nonlinearity and the parameters of the Gaussian approximation are then re-estimated. For some problems, this filter has been shown to give better performance than a standard EKF since it better approximates the nonlinearity; the parameters of the Gaussian approximation are improved.

However, the EKF always approximates $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ to be Gaussian. If the true density is non-Gaussian (eg. if it is bi-modal or heavily skewed) then a Gaussian can never describe it well. In such cases, approximate grid-based filters and particle filters will yield an improvement in performance in comparison to that of an EKF [2].

4 Particle Filtering Methods

4.1 The Sequential Importance Sampling (SIS) Algorithm

The Sequential Importance Sampling (SIS) algorithm is a Monte Carlo (MC) method that forms the basis for most sequential Monte Carlo filters developed over the past decades – see [13, 14]. This sequential Monte Carlo (SMC) approach is known variously as bootstrap filtering [17], the condensation algorithm [25], particle filtering [6], interacting particle approximations [10, 11] and survival of the fittest [21]. It is a technique for implementing a recursive Bayesian filter by Monte Carlo simulations. The key idea is to represent the required posterior density function by a set of random samples with associated weights and to compute estimates based on these samples and weights. As the number of samples becomes very large, this Monte Carlo characterisation becomes an equivalent representation to the usual functional description of the posterior pdf, and the SIS filter approaches the optimal Bayesian estimate.

In order to develop the details of the algorithm, let $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^{N_s}$ denote a *Random Measure* that characterises the posterior pdf $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$, where $\{\mathbf{x}_{0:k}^i, i = 0, \ldots, N_s\}$ is a set of support points with associated weights $\{w_k^i, i = 1, \ldots, N_s\}$ and $\mathbf{x}_{0:k} = \{\mathbf{x}_j, j = 0, \ldots, k\}$ is the set of all states up to time k. The weights are normalised such that $\sum_i w_k^i = 1$. Then, the posterior density at k can be approximated as

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i), \quad (14)$$

We therefore have a discrete weighted approximation to the true posterior, $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. The weights are chosen using the principle of *Importance Sampling* [3, 12]. This principle relies on the following: Suppose $p(x) \propto \pi(x)$ is a probability density from which it is difficult to draw samples, but for which $\pi(x)$ can be evaluated (and so $p(x)$ up to proportionality). Also, let $x^i \sim q(x)$, $i = 1, \ldots, N_s$ be samples that are easily generated from a proposal $q(\cdot)$, called an *Importance density*. Then, a weighted approximation to the density $p(\cdot)$ is given by

$$p(x) \approx \sum_{i=1}^{N_s} w^i \delta(x - x^i) \quad (15)$$

where

$$w^i \propto \frac{\pi(x^i)}{q(x^i)} \quad (16)$$

is the normalised weight of the i^{th} particle.

So, if the samples, $\mathbf{x}^i_{0:k}$, were drawn from an importance density, $q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ then the weights in (14) are defined by (16) to be

$$w^i_k \propto \frac{p(\mathbf{x}^i_{0:k}|\mathbf{z}_{1:k})}{q(\mathbf{x}^i_{0:k}|\mathbf{z}_{1:k})} \tag{17}$$

Returning to the sequential case, at each iteration, one could have samples constituting an approximation to $p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})$, and want to approximate $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ with a new set of samples. If the importance density is chosen to factorise such that

$$q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \tag{18}$$

then one can obtain samples $\mathbf{x}^i_{0:k} \sim q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ by augmenting each of the existing samples $\mathbf{x}^i_{0:k-1} \sim q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})$ with the new state $\mathbf{x}^i_k \sim q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$. To derive the weight update equation, $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ is first expressed in terms of $p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})$, $p(\mathbf{z}_k|\mathbf{x}_k)$ and $p(\mathbf{x}_k|\mathbf{x}_{k-1})$.

$$\begin{aligned}
p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) &= \frac{p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})\, p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \\
&= \frac{p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1})\, p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1})\, p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \\
&= \frac{p(\mathbf{z}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \\
&\propto p(\mathbf{z}_k|\mathbf{x}_k)\, p(\mathbf{x}_k|\mathbf{x}_{k-1})\, p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})
\end{aligned} \tag{19}$$

By substituting (18) and (19) into (17), the weight update equation can then be shown to be

$$\begin{aligned}
w^i_k &\propto \frac{p(\mathbf{z}_k|\mathbf{x}^i_k)\, p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})\, p(\mathbf{x}^i_{0:k-1}|\mathbf{z}_{1:k-1})}{q(\mathbf{x}^i_k|\mathbf{x}^i_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}^i_{0:k-1}|\mathbf{z}_{1:k-1})} \\
&= w^i_{k-1} \frac{p(\mathbf{z}_k|\mathbf{x}^i_k) p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})}{q(\mathbf{x}^i_k|\mathbf{x}^i_{0:k-1}, \mathbf{z}_{1:k})}
\end{aligned} \tag{20}$$

Furthermore, if $q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$, then the importance density becomes only dependent on the $\mathbf{x}_{k-1}$ and $\mathbf{z}_k$. This is particularly useful in the common case when only a filtered estimate of $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is required at each time step. From this point on, we shall assume such a case, except when explicitly stated otherwise. In such scenarios, only $\mathbf{x}^i_k$ need be stored, and so one can discard the path, $\mathbf{x}^i_{0:k-1}$, and history of observations, $\mathbf{z}_{1:k-1}$. The modified weight is then

$$w^i_k \propto w^i_{k-1} \frac{p(\mathbf{z}_k|\mathbf{x}^i_k) p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})}{q(\mathbf{x}^i_k|\mathbf{x}^i_{k-1}, \mathbf{z}_k)} \tag{21}$$

and the posterior filtered density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ can be approximated as

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w^i_k \delta(\mathbf{x}_k - \mathbf{x}^i_k) \tag{22}$$

where the weights are defined in (21). It can be shown that as $N_s \to \infty$ the approximation (22) approaches the true posterior density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$.

The SIS algorithm thus consists of recursive propagation of the weights and support points as each measurement is received sequentially. A pseudo-code description of this algorithm is given by algorithm 1.

ALGORITHM 1: SIS PARTICLE FILTER

$[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s}] = \text{SIS}\ [\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, \mathbf{z}_k]$

- FOR $i = 1 : N_s$
  - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$
  - Assign the particle a weight, w_k^i, according to (21)
- END FOR

4.1.1 Degeneracy Problem

A common problem with the SIS particle filter is the degeneracy phenomenon, where after a few iterations, all but one particle will have negligible weight. It has been shown [12] that the variance of the importance weights can only increase over time, and thus it is impossible to avoid the degeneracy phenomenon. This degeneracy implies that a large computational effort is devoted to updating particles whose contribution to the approximation to $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is almost zero. A suitable measure of degeneracy of the algorithm is the effective sample size N_{eff} introduced in [3] and [24], and defined as

$$N_{eff} = \frac{N_s}{1 + \text{Var}(w_k^{*i})}, \tag{23}$$

where $w_k^{*i} = p(\mathbf{x}_k^i|\mathbf{z}_{1:k})/q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$ is referred to as the "true weight". This cannot be evaluated exactly, but an estimate $\widehat{N_{eff}}$ of N_{eff} can be obtained by

$$\widehat{N_{eff}} = \frac{1}{\sum_{i=1}^{N_s}(w_k^i)^2} \tag{24}$$

where w_k^i is the normalised weight obtained using (20). Notice that $N_{eff} \leq N_s$, and small N_{eff} indicates severe degeneracy. Clearly, the degeneracy problem is an undesirable effect in particle filters. The brute force approach to reducing its effect is to use a very large N_s. This is often impractical, and so we rely on two other methods: a) Good choice of importance density, and b) Use of resampling. These are described next.

4.1.2 Good Choice of Importance Density

The first method involves choosing the importance density, $q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$, to minimise $\text{Var}(w_k^{*i})$ so that N_{eff} is maximised. The optimal importance density function which minimises the variance of the true weights, w_k^{*i}, conditioned upon $\mathbf{x}_{k-1}^i$ and $\mathbf{z}_k$ has been shown [12] to be

$$\begin{aligned} q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)_{opt} &= p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k) \\ &= \frac{p\left(\mathbf{z}_k|\mathbf{x}_k, \mathbf{x}_{k-1}^i\right) p\left(\mathbf{x}_k|\mathbf{x}_{k-1}^i\right)}{p\left(\mathbf{z}_k|\mathbf{x}_{k-1}^i\right)}. \end{aligned} \tag{25}$$

Substitution of (25) into (21) yields

$$\begin{aligned} w_k^i &\propto w_{k-1}^i p\left(\mathbf{z}_k|\mathbf{x}_{k-1}^i\right) \\ &= w_{k-1}^i \int p\left(\mathbf{z}_k|\mathbf{x}_k'\right) p\left(\mathbf{x}_k'|\mathbf{x}_{k-1}^i\right) d\mathbf{x}_k'. \end{aligned} \tag{26}$$

This choice of importance density is optimal since for a given $\mathbf{x}_{k-1}^i$, w_k^i takes the same value whatever sample is drawn from $q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)_{opt}$. Hence, conditional on $\mathbf{x}_{k-1}^i$, $\mathrm{Var}(w_k^{*i}) = 0$. This is the variance of the different w_k^i resulting from different sampled $\mathbf{x}_k^i$.

This optimal importance density suffers from two major drawbacks. It requires the ability to sample from $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$ and to evaluate the integral over the new state. In the general case it may not be straightforward to do either of these things. There are two cases when use of the optimal importance density is possible.

The first case is when $\mathbf{x}_k$ is a member of a finite set. In such cases, the integral in (25) becomes a sum and sampling from $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$ is possible. An example of an application when $\mathbf{x}_k$ is a member of a finite set is a Jump-Markov Linear System for tracking maneuvering targets [15].

Analytic evaluation is possible for a second class of models for which $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$ is Gaussian [12, 9]. This can occur if the dynamics are nonlinear and the measurements linear. Such a system is given by

$$\begin{array}{rclrcl} \mathbf{x}_k &=& \mathbf{f}_k(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1}, & \mathbf{v}_{k-1} &\sim& \mathcal{N}(\mathbf{v}_{k-1}; \mathbf{0}_{n_v \times 1}, Q_{k-1}) \\ \mathbf{z}_k &=& H_k \mathbf{x}_k + \mathbf{n}_k, & \mathbf{n}_k &\sim& \mathcal{N}(\mathbf{n}_k; \mathbf{0}_{n_n \times 1}, R_k), \end{array} \tag{27}$$

where $\mathbf{f}_k : \Re^{n_x} \to \Re^{n_x}$ is a nonlinear function, $H_k \in \Re^{n_z \times n_x}$ is an observation matrix, and $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are mutually independent i.i.d Gaussian sequences with $Q_{k-1} > \mathbf{0}$ and $R_k > \mathbf{0}$. Defining

$$\Sigma_k^{-1} = Q_{k-1}^{-1} + H_k^T R_k^{-1} H_k \tag{28}$$

$$\mathbf{m}_k = \Sigma_k \left(Q_{k-1}^{-1} \mathbf{f}_k(\mathbf{x}_{k-1}) + H_k^T R_k^{-1} \mathbf{z}_k \right), \tag{29}$$

one obtains

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_k, \Sigma_k) \tag{30}$$

and

$$p(\mathbf{z}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{z}_k; H_k \mathbf{f}_k(\mathbf{x}_{k-1}), Q_{k-1} + H_k R_k H_k^T). \tag{31}$$

For many other models, such analytic evaluations are not possible. However, it is possible to construct suboptimal approximations to the optimal importance density by using local linearisation techniques [12]. Such linearisations use an importance density that is a Gaussian approximation to $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$. Another approach is to estimate a Gaussian approximation to $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$ using the unscented transform [30]. The authors' opinion is that the additional computational cost of using such an importance density is often more than offset by a reduction in the number of samples required to achieve a certain level of performance.

Finally, it is often convenient to choose the importance density to be the prior.

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}^i) \tag{32}$$

Substitution of (32) into (21) then yields

$$w_k^i \propto w_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i). \tag{33}$$

This would seem to be the most common choice of importance density since it is intuitive and simple to implement. However, there are a plethora of other densities that can be used and the choice is the crucial design step for a particle filter.

4.1.3 Resampling

The second method by which the effects of degeneracy can be reduced is to use resampling whenever a significant degeneracy is observed (ie. when N_{eff} falls below some threshold, N_T). The basic idea of resampling is to eliminate particles which have small weights and to concentrate on particles with large

weights. The resampling step involves generating a new set $\{\mathbf{x}_k^{i^*}\}_{i=1}^{N_s}$ by resampling (with replacement) N_s times from an approximate discrete representation of $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ given by

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{34}$$

so that $Pr(\mathbf{x}_k^{i^*} = \mathbf{x}_k^j) = w_k^j$. The resulting sample is in fact an i.i.d sample from the discrete density (34), and so the weights are now reset to $w_k^i = 1/N_s$. It is possible to implement this resampling procedure in $O(N_s)$ operations by sampling N_s ordered uniforms using an algorithm based on order statistics [29, 6]. Note that other efficient (in terms of reduced MC variation) resampling schemes such as stratified sampling and residual sampling [24] may be applied as alternatives to this algorithm. Systematic resampling [22] is the scheme preferred by the authors (since it is simple to implement, takes $O(N_s)$ time and minimises the MC variation) and its operation is described in algorithm 2, where $\mathrm{U}[a,b]$ is the Uniform distribution on the interval $[a,b]$. For each resampled particle $\mathbf{x}_k^{j^*}$, this resampling algorithm also stores the index of its parent, denoted by i^j. This may appear unnecessary here and is, but it proves useful in section 4.2.2.

A generic particle filter is then as described by algorithm 3.

ALGORITHM 2: RESAMPLING ALGORITHM

$[\{\mathbf{x}_k^{j^*}, w_k^j, i^j\}_{j=1}^{N_s}] = \text{RESAMPLE } [\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s}]$

- Initialise the CDF: $c_1 = 0$
- FOR $i = 2 : N_s$
  - Construct CDF: $c_i = c_{i-1} + w_k^i$
- END FOR
- Start at the bottom of the CDF: $i = 1$
- Draw a starting point: $u_1 \sim \mathrm{U}\left[0, N_s^{-1}\right]$
- FOR $j = 1 : N_s$
  - Move along the CDF: $u_j = u_1 + N_s^{-1}(j-1)$
  - WHILE $u_j > c_i$
    - $i = i + 1$
  - END WHILE
  - Assign sample: $\mathbf{x}_k^{j^*} = \mathbf{x}_k^i$
  - Assign weight: $w_k^j = N_s^{-1}$
  - Assign parent: $i^j = i$
- END FOR

ALGORITHM 3: GENERIC PARTICLE FILTER

$[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s}] = \text{PF } [\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, \mathbf{z}_k]$

- FOR $i = 1 : N_s$
  - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$
  - Assign the particle a weight, w_k^i, according to (21)
- END FOR
- Calculate total weight: $t = \text{SUM } [\{w_k^i\}_{i=1}^{N_s}]$
- FOR $i = 1 : N_s$
  - Normalise: $w_k^i = t^{-1} w_k^i$
- END FOR
- Calculate $\widehat{N_{eff}}$ using (24)
- IF $\widehat{N_{eff}} < N_T$
  - Resample using algorithm 2: $[\{\mathbf{x}_k^i, w_k^i, -\}_{i=1}^{N_s}] = \text{RESAMPLE } [\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s}]$
- END IF

Although the resampling step reduces the effects of the degeneracy problem, it introduces other practical problems. First, it limits the opportunity to parallelize since all the particles must be combined. Second, the particles which have high weights w_k^i are statistically selected many times. This leads to a loss of diversity among the particles as the resultant sample will contain many repeated points. This problem, known as *sample impoverishment*, is severe in the case of small process noise. In fact, for the case of very small process noise all particles will collapse to a single point within a few iterations[1]. Thirdly, since the diversity of the paths of the particles is reduced, any smoothed estimates based on the particles' paths degenerate[2]. Schemes exist to counteract this effect. One approach considers the states for the particles to be pre-determined by the forward filter and then obtains the smoothed estimates by re-calculating the particles' weights via a recursion from the final to the first time step [16]. Another approach is to use MCMC [5].

There have been some systematic techniques proposed recently to solve the problem of sample impoverishment. One such technique is the *resample-move* algorithm [18]. Although this technique draws conceptually on the same technologies of importance sampling-resampling and MCMC sampling, it avoids sample impoverishment. It does this in a rigorous manner that ensures the particles asymptotically approximate samples from the posterior and so is the method of choice of the authors. An alternative solution to the same problem is *regularisation* [26], which is discussed in Section 4.2.3. This approach is frequently found to improve performance despite a less rigorous derivation and is included here in preference to the resample-move algorithm since its use is so widespread.

[1] If the process noise is zero then using a particle filter is not entirely appropriate. Particle filtering is a method well suited to the estimation of dynamic states. If static states, which can be regarded as parameters, need to be estimated then alternative approaches are necessary [7, 23].

[2] Since the particles actually represent paths through the state space, by storing the trajectory taken by each particle, smoothed estimates of the state can be obtained [4].

4.1.4 Techniques for Circumventing the Use of a Sub-Optimal Importance Density

It is often the case that a good importance density is not available. For example, if the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is used as the importance density and is a much broader distribution than the likelihood, $p(\mathbf{z}_k|\mathbf{x}_k)$, then only a few particles will have a high weight. Methods exist for encouraging the particles to be in the right place; the use of bridging densities [8] and progressive correction [27] both introduce intermediate distributions between the prior and likelihood. The particles are then re-weighted according to these intermediate distributions and resampled. This "herds" the particles into the right part of the state space.

Another approach known as partitioned sampling [25] is useful if the likelihood is very peaked, but can be factorised into a number of broader distributions. Typically, this occurs because each of the partitioned distributions are functions of some (not all) of the states. By treating each of these partitioned distributions in turn and resampling on the basis of each such partitioned distribution, the particles are again herded towards the peaked likelihood.

4.2 Other Related Particle Filters

The Sequential Importance Sampling algorithm presented in Section V-A forms the basis for most particle filters that have been developed so far. The various versions of particle filters proposed in the literature can be regarded as special cases of this general SIS algorithm. These special cases can be derived from the SIS algorithm by an appropriate choice of importance sampling density and/or modification of the resampling step. Below, we present three particle filters proposed in the literature and show how these may be derived from the SIS algorithm. The particle filters considered are (i) Sampling Importance Resampling (SIR) filter (ii) Auxiliary Sampling Importance Resampling (ASIR) filter, and (iii) Regularised Particle filter (RPF).

4.2.1 Sampling Importance Resampling Filter

The Sampling Importance Resampling (SIR) filter proposed in [17] is a Monte Carlo method that can be applied to recursive Bayesian filtering problems. The assumptions required to use the SIR filter are very weak. The state dynamics and measurement functions, $\mathbf{f}_k(\cdot,\cdot)$ and $\mathbf{h}_k(\cdot,\cdot)$ in (1) and (2) respectively, need to be known, it is required to be able to sample realisations from the process noise distribution of $\mathbf{v}_{k-1}$ and from the prior. Finally, the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ needs to be available for pointwise evaluation (at least up to proportionality). The SIR algorithm can be easily derived from the SIS algorithm by an appropriate choice of: (i) The importance density: $q(\mathbf{x}_k|\mathbf{x}^i_{k-1},\mathbf{z}_{1:k})$ is chosen to be the prior density $p(\mathbf{x}_k|\mathbf{x}^i_{k-1})$, and (ii) Resampling step: to be applied at every time index.

The above choice of importance density implies that we need samples from $p(\mathbf{x}_k|\mathbf{x}^i_{k-1})$. A sample $\mathbf{x}^i_k \sim p(\mathbf{x}_k|\mathbf{x}^i_{k-1})$ can be generated by first generating a process noise sample $\mathbf{v}^i_{k-1} \sim p_v(\mathbf{v}_{k-1})$ and setting $\mathbf{x}^i_k = \mathbf{f}_k(\mathbf{x}^i_{k-1},\mathbf{v}^i_{k-1})$, where $p_v(\cdot)$ is the pdf of $\mathbf{v}_{k-1}$. For this particular choice of importance density, it is evident that the weights are given by

$$w^i_k \propto w^i_{k-1} p(\mathbf{z}_k|\mathbf{x}^i_k). \tag{35}$$

However, noting that resampling is applied at every time index, we have $w^i_{k-1} = 1/N \ \forall i$ and so

$$w^i_k \propto p(\mathbf{z}_k|\mathbf{x}^i_k). \tag{36}$$

The weights given by the proportionality in (36) are normalised before the resampling stage. An iteration of the algorithm is then described by algorithm 4.

ALGORITHM 4: SIR PARTICLE FILTER

$[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s}] = \text{SIR}\ [\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, \mathbf{z}_k]$

- FOR $i = 1 : N_s$
  - Draw $\mathbf{x}_k^i \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$
  - Calculate $w_k^i = p(\mathbf{z}_k|\mathbf{x}_k^i)$
- END FOR
- Calculate total weight: $t = \text{SUM}\ [\{w_k^i\}_{i=1}^{N_s}]$
- FOR $i = 1 : N_s$
  - Normalise: $w_k^i = t^{-1} w_k^i$
- END FOR
- Resample using algorithm 2: $[\{\mathbf{x}_k^i, w_k^i, -\}_{i=1}^{N_s}] = \text{RESAMPLE}\ [\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s}]$

As the importance sampling density for the SIR filter is independent of measurement $\mathbf{z}_k$, the state space is explored without any knowledge of the observations. Therefore, this filter can be inefficient and is sensitive to outliers. Furthermore, as resampling is applied at every iteration, this can result in rapid loss of diversity in particles. However, the SIR method does have the advantage that the importance weights are easily evaluated and the importance density can be easily sampled.

4.2.2 Auxiliary Sampling Importance Resampling filter

The Auxiliary Sampling Importance Resampling (ASIR) filter was introduced by Pitt & Shephard [28] as a variant of the standard SIR filter. This filter can be derived from the SIS framework by introducing an importance density $q(\mathbf{x}_k, i|\mathbf{z}_{1:k})$ which samples the pair $\{\mathbf{x}_k^j, i^j\}_{j=1}^{N_s}$, where i^j refers to the index of the particle at $k-1$.

By applying Bayes' rule, a proportionality can be derived for $p(\mathbf{x}_k, i|\mathbf{z}_{1:k})$ as

$$\begin{aligned} p(\mathbf{x}_k, i|\mathbf{z}_{1:k}) &\propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k, i|\mathbf{z}_{1:k-1}) \\ &= p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|i, \mathbf{z}_{1:k-1})p(i|\mathbf{z}_{1:k-1}) \\ &= p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)w_{k-1}^i \end{aligned} \tag{37}$$

The ASIR filter operates by obtaining a sample from the joint density $p(\mathbf{x}_k, i|\mathbf{z}_{1:k})$, and then omitting the indices i in the pair $(\mathbf{x}_k, i)$ to produce a sample $\{\mathbf{x}_k^j\}_{j=1}^{N_s}$ from the marginalised density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. The importance density used to draw the sample $\{\mathbf{x}_k^j, i^j\}_{j=1}^{N_s}$ is defined to satisfy the proportionality

$$q(\mathbf{x}_k, i|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mu_k^i)p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)w_{k-1}^i \tag{38}$$

where μ_k^i is some characterisation of $\mathbf{x}_k$, given $\mathbf{x}_{k-1}^i$. This could be the mean, in which case $\mu_k^i = \mathbb{E}[\mathbf{x}_k|\mathbf{x}_{k-1}^i]$, or a sample, $\mu_k^i \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$. By writing

$$q(\mathbf{x}_k, i|\mathbf{z}_{1:k}) = q(i|\mathbf{z}_{1:k})q(\mathbf{x}_k|i, \mathbf{z}_{1:k}), \tag{39}$$

and defining

$$q(\mathbf{x}_k|i,\mathbf{z}_{1:k}) \triangleq p(\mathbf{x}_k|\mathbf{x}_{k-1}^i), \tag{40}$$

it follows from (38) that

$$q(i|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mu_k^i)w_{k-1}^i. \tag{41}$$

The sample, $\{\mathbf{x}_k^j, i^j\}_{j=1}^{N_s}$, is then assigned a weight proportional to the ratio of the RHS of (37) to (38):

$$\begin{aligned} w_k^j &\propto w_{k-1}^{i^j} \frac{p(\mathbf{z}_k|\mathbf{x}_k^j)p(\mathbf{x}_k^j|\mathbf{x}_{k-1}^{i^j})}{q(\mathbf{x}_k^j, i^j|\mathbf{z}_{1:k})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k^j)}{p(\mathbf{z}_k|\mu_k^{i^j})} \end{aligned} \tag{42}$$

The algorithm then becomes that described by algorithm 5.

ALGORITHM 5: AUXILIARY PARTICLE FILTER

$[\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s}] = \text{APF}\ [\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, \mathbf{z}_k]$

- FOR $i = 1 : N_s$
  - Calculate μ_k^i
  - Calculate $w_k^i = q(i|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mu_k^i)w_{k-1}^i$.
- END FOR
- Calculate total weight: $t = \text{SUM}\ [\{w_k^i\}_{i=1}^{N_s}]$
- FOR $i = 1 : N_s$
  - Normalise: $w_k^i = t^{-1}w_k^i$
- END FOR
- Resample using algorithm 2: $[\{-,-,i^j\}_{j=1}^{N_s}] = \text{RESAMPLE}\ [\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s}]$
- FOR $j = 1 : N_s$
  - Draw $\mathbf{x}_k^j \sim q(\mathbf{x}_k|i^j,\mathbf{z}_{1:k}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}^{i^j})$, as in the SIR filter.
  - Assign weight w_k^j using (42)
- END FOR
- Calculate total weight: $t = \text{SUM}\ [\{w_k^i\}_{i=1}^{N_s}]$
- FOR $i = 1 : N_s$
  - Normalise: $w_k^i = t^{-1}w_k^i$
- END FOR

Although unnecessary, the original ASIR filter as proposed in [28] consisted of one more step, namely a resampling stage, to produce an i.i.d sample $\{\mathbf{x}_k^j, i^j\}_{j=1}^{N_s}$ with equal weights.

Compared to the SIR filter, the advantage of the ASIR filter is that it naturally generates points from the sample at $k-1$ which, conditioned on the current measurement, are most likely to be close to the true state. ASIR can be viewed as resampling at the previous time step, based on some point estimates, μ_k^i, that characterise $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$. If the process noise is small, so $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ is well characterised by μ_k^i, then ASIR is often not so sensitive to outliers as SIR, and the weights w_k^i are more even. However, if the process noise is large, a single point does not characterise $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ well and ASIR resamples based on a poor approximation of $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$. In such scenarios, the use of ASIR then degrades performance.

4.2.3 The Regularised Particle Filter

Recall that resampling was suggested in section 4.2.1, as a method to reduce the degeneracy problem which is prevalent in particle filters. However, it was pointed out that resampling in turn introduced other problems, and in particular, the problem of loss of diversity among the particles. This arises due to the fact that in the resampling stage, samples are drawn from a discrete distribution rather than a continuous one. If this problem is not addressed properly, it may lead to "particle collapse", where all N_s particles occupy the same point in the state space, giving a poor representation of the posterior density. A modified particle filter known as the Regularised Particle Filter (RPF) was proposed [26] as a potential solution to the above problem.

The Regularised Particle filter is identical to the SIR filter except for the resampling stage. The RPF resamples from a continuous approximation of the posterior density $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ while the SIR resamples from the discrete approximation (34). Specifically, in the RPF, samples are drawn from the approximation

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i K_h(\mathbf{x}_k - \mathbf{x}_k^i) \tag{43}$$

where

$$K_h(\mathbf{x}) = \frac{1}{h^{n_x}} K\left(\frac{\mathbf{x}}{h}\right) \tag{44}$$

is the re-scaled Kernel density $K(\cdot)$, $h > 0$ is the Kernel bandwidth (a scalar parameter), n_x is the dimension of the state vector $\mathbf{x}$, and w_k^i, $i = 1, \ldots, N_s$, are normalised weights. The Kernel density is a symmetric probability density function such that

$$\int \mathbf{x} K(\mathbf{x}) d\mathbf{x} = \mathbf{0}, \quad \int \|\mathbf{x}\|^2 K(\mathbf{x}) d\mathbf{x} < \infty$$

The Kernel $K(\cdot)$ and bandwidth h are chosen to minimise the Mean Integrated Square Error (MISE) between the true posterior density and the corresponding regularised empirical representation in (43), defined as

$$\text{MISE}(\hat{p}) = \mathbb{E}\left[\int \left[\hat{p}(\mathbf{x}_k|\mathbf{z}_{1:k}) - p(\mathbf{x}_k|\mathbf{z}_{1:k})\right]^2 d\mathbf{x}_k\right] \tag{45}$$

where $\hat{p}(\cdot|\cdot)$ denotes the approximation to $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ given by the RHS of (43). In the special case of an equally weighted sample, the optimal choice of the Kernel is the Epanechnikov Kernel [26],

$$K_{opt} = \begin{cases} \frac{n_x+2}{2c_{n_x}}(1 - \|\mathbf{x}\|^2) & \text{if } \|\mathbf{x}\| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{46}$$

where c_{n_x} is the volume of the unit hypersphere in $\Re^{n_x}$. Furthermore, when the underlying density is Gaussian with a unit covariance matrix, the optimal choice for the bandwidth is [26]

$$h_{opt} = A N_s^{\frac{1}{n_x+4}} \tag{47}$$

$$A = [8c_{n_x}^{-1}(n_x+4)(2\sqrt{\pi})^{n_x}]^{\frac{1}{n_x+4}} \tag{48}$$

Though the results of (46) and (47)-(48) are optimal only in the special case of equally weighted particles and underlying Gaussian density, these results can still be used in the general case to obtain a suboptimal

filter. One iteration of the RPF is described by algorithm 6. The RPF only differs from the generic particle filter described by algorithm 3 as a result of the addition of the regularisation steps when conducting the resampling. Note also that the calculation of the empirical covariance matrix S_k is carried out prior to the resampling and so is a function of both the $\mathbf{x}_k^i$ and w_k^i. This is done since the accuracy of any estimate of a function of the distribution can only decrease as a result of the resampling - if quantities such as the mean and covariance of the samples are to be output then these should be calculated prior to resampling.

ALGORITHM 6: REGULARISED PARTICLE FILTER

$[\{\mathbf{x}_k^{i\,*}, w_k^i\}_{i=1}^{N_s}] = \text{RPF}\ [\{\mathbf{x}_{k-1}^i, w_{k-1}^i\}_{i=1}^{N_s}, \mathbf{z}_k]$

- FOR $i = 1 : N_s$
  - Draw $\mathbf{x}_k^i \sim q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$
  - Assign the particle a weight, w_k^i, according to (21)
- END FOR
- Calculate total weight: $t = \text{SUM}\ [\{w_k^i\}_{i=1}^{N_s}]$
- FOR $i = 1 : N_s$
  - Normalise: $w_k^i = t^{-1} w_k^i$
- END FOR
- Calculate $\widehat{N_{eff}}$ using (24)
- IF $\widehat{N_{eff}} < N_T$
  - Calculate the empirical covariance matrix S_k of $\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s}$
  - Compute $\mathbf{D}_k$ such that $\mathbf{D}_k \mathbf{D}_k^T = S_k$.
  - Resample using algorithm 2: $[\{\mathbf{x}_k^i, w_k^i, -\}_{i=1}^{N_s}] = \text{RESAMPLE}\ [\{\mathbf{x}_k^i, w_k^i\}_{i=1}^{N_s}]$
  - FOR $i = 1 : N_s$
    - Draw $\epsilon^i \sim K$ from the Epanechnikov Kernel
    - $\mathbf{x}_k^{i\,*} = \mathbf{x}_k^i + h_{opt} \mathbf{D}_k \epsilon^i$
  - END FOR
- END IF

By following the above procedure, we generate an i.i.d random sample $\{\mathbf{x}_k^{i\,*}\}_{i=1}^{N_s}$ drawn from (43).

In terms of complexity, the RPF is comparable to SIR since it only requires N_s additional generations from the Kernel $K(\cdot)$ at each time step. The RPF has the theoretic disadvantage that the samples are no longer guaranteed to asymptotically approximate those from the posterior. In practical scenarios, the RPF performance is better than the SIR in cases where sample impoverishment is severe, for example when the process noise is small.

5 Conclusions

For a particular problem, if the assumptions of the Kalman filter hold then no other algorithm can out-perform it. However, in a variety of real scenarios, the assumptions do not hold and approximate techniques must be employed.

The extended Kalman filter approximates the models used for the dynamics and measurement process, in order to be able to approximate the probability density by a Gaussian. Particle filtering approximates the density directly as a finite number of samples. A number of different types of particle filter exist and some have been shown to outperform others when used for particular applications. However, when designing a particle filter for a particular application, it is the choice of importance density that is critical.

References

[1] S Arulampalam, S R Maskell, N J Gordon & T Clapp, A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking, *To appear: IEEE Transactions on Signal Processing.*

[2] S Arulampalam & B Ristic, Comparison of the Particle Filter with Range Parameterised and Modified Polar EKF's for Angle-Only Tracking. *Signal and Data Processing of Small Targets 2000*, SPIE Volume 4048, pp.288-299.

[3] N Bergman, Recursive Bayesian Estimation: Navigation and Tracking Applications. 1999, PhD Thesis, Linkoping University, Sweden.

[4] N Bergman, A Doucet & N Gordon, Optimal Estimation and Cramer-Rao Bounds for Partial Non-Gaussian State Space Models. *Ann. Inst. Statist. Math.*, 2001, Volume 53(1), pp 97-112.

[5] B P Carlin, N G Polson & D S Stoffer, A Monte Carlo approach to nonnormal and nonlinear state-space modelling, *Journal of the American Statistical Association*, 1992, 87(418), 493-500.

[6] J Carpenter, P Clifford & P Fearnhead, Improved Particle Filter for Nonlinear Problems, In *IEE Proceedings on Radar and Sonar Navigation*, 1999.

[7] T Clapp, Statistical Methods for the Processing of Communications Data, PhD thesis, Department of Engineering, University of Cambridge, 2000.

[8] T Clapp & S Godsill, Improvement strategies for Monte Carlo particle filters. *in Sequential Monte Carlo Methods in Practice, eds A Doucet, J F G de Freitas and N J Gordon*, 2001, Springer-Verlag: New York.

[9] P Del Moral, Measure Valued Processes and Interacting Particle Systems. Application to Nonlinear Filtering Problems. *Annals of Applied Probability*, 1998, Volume 8, No2, pp 438-495.

[10] D Crisan, P Del Moral & T J Lyons, Nonlinear Filtering Uing Branching and Interacting Particle Systems. *Markov Processes and Related Fields*, 1999, Volume 5, No 3, pp 293-319.

[11] P Del Moral, Nonlinear Filtering: Interacting Particle Solution, *Markov Processes and Related Fields*, Volume 2, No 4, pp 555-580.

[12] A Doucet, On Sequential Monte Carlo Methods for Bayesian Filtering. 1998, Technical Report, University of Cambridge, UK, Department of Engineering.

[13] A Doucet, J F G de Freitas & N J Gordon, An introduction to sequential Monte Carlo methods, *in Sequential Monte Carlo Methods in Practice, eds A Doucet, J F G de Freitas and N J Gordon*, 2001, Springer-Verlag: New York.

[14] A Doucet, S Godsill & C Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing*, 10(3), 197-208.

[15] A Doucet, N Gordon & V Krishnamurthy, Particle Filters for State Estimation of Jump Markov Linear Systems. *IEEE Transactrions on Signal Processing*, March 2001, Volume 49 , No 3, pp 613-624.

[16] S Godsill, A Doucet & M West, Methodology for Monte Carlo Smoothing with Application to Time-Varying Autoregressions. *Proc. International Symposium on Frontiers of Time Series Modelling*, 2000.

[17] N Gordon, D Salmond & A F M Smith, Novel Approach to Nonlinear and Non-Gaussian Bayesian State Estimation. *IEE Proceedings-F*, 1993, Volume 140, pp.107-113.

[18] W R Gilks & C Berzuini, Following a Moving Target – Monte Carlo inference for Dynamic Baysian Models. *Journal of the Royal Statistical Society*, B, 2001, Volume 63, pp. 127-146.

[19] A H Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press: New York, 1970.

[20] S Julier, A Skewed Approach to Filtering. *Signal and Data Processing of Small Targets 1998*, SPIE Volume 3373, pp.271-282.

[21] K Kanazawa, D Koller & S J Russell, Stochastic Simulation Algorithms for Dynamic Probabilistic Networks. In *Proceedings of the Eleventh Annual Conference on Uncertainty in AI (UAI '95)*, 1995, pp 346-351.

[22] G Kitagawa, Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal Of Computational and Graphical Statistics*, 1996, Volume 5(1), pp.1-25.

[23] J Liu & M West, Combined parameter and state estimation in simulation-based filtering, *in Sequential Monte Carlo Methods in Practice, eds A Doucet, J F G de Freitas and N J Gordon*, 2001, Springer-Verlag: New York.

[24] J S Liu & R Chen, Sequential Monte Carlo Methods for Dynamical Systems. *Journal of the American Statistical Association*, 1998, Volume 93, pp.1032-1044.

[25] J MacCormick & A Blake, A Probabilistic Exclusion Principle for Tracking Multiple Objects. *Proc Int. Conf. Computer Vision*, 1999, pp.572-578.

[26] C Musso, N Oudjane & F LeGland, Improving Regularised Particle Filters. *in: Sequential Monte Carlo Methods in Practice, eds A Doucet, J F G de Freitas and N J Gordon.* 2001, Springer-Verlag, New York.

[27] N Oudjane & C Musso, Progressive Correction for Regularized Particle Filters. *Proc. 3rd International Conference on Information Fusion*, 2000.

[28] M Pitt & N Shephard, Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, 1999, Volume 94(446), pp.590-599.

[29] B Ripley, *Stochastic Simulation*, 1987, Wiley, New York.

[30] R van der Merwe, A Doucet, J F G de Freitas & E Wan, The Unscented Particle Filter. In *Advances in Neural Information Processing Systems (NIPS13)*, December 2000.

[31] E A Wan & R Van der Merwe, The Unscented Kalman Filter for Nonlinear Estimation. In *Proceedings of Symposium 2001 on Adaptive Systems for Signal Processing, Communications and Control*, Lake Louise, Alberta, Canada, Oct 2000.

[32] E A Wan & R Van der Merwe, To appear in *Kalman Filtering and Neural Networks*, Chapter 7: The Unscented Kalman Filter, Wiley Publishing, 2001.