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

Sanjeev Arulampalam, Simon Maskell, Neil Gordon and Tim Clapp

Abstract—**Increasingly for many application areas it is becoming important to include elements of non-linearity and non-Gaussianity, in order to model accurately the underlying dynamics of a physical system. Moreover, it is typically crucial to process data on-line as it arrives, both from the point of view of storage costs and also for rapid adaptation to changing signal characteristics. In this paper we review both optimal and suboptimal Bayesian algorithms for non-linear/non-Gaussian tracking problems, with a focus on Particle filters. Particle filters are sequential Monte Carlo methods based upon point mass (or 'particle') representations of probability densities, which can be applied to any state space model, and which generalise the traditional Kalman filtering methods. Several variants of the particle filter such as SIR, ASIR, and RPF, are introduced within a generic framework of the Sequential Importance Sampling (SIS) algorithm. These are discussed and compared with the standard EKF through an illustrative example.**

Keywords – Particle Filters, Sequential Monte Carlo, Bayesian, Non-linear/Non-Gaussian, Tracking.

I. Introduction

Many problems in science require estimation of the state of a system that changes over time using a sequence of noisy measurements made on the system. In this article we shall concentrate on the state space approach to modelling dynamic systems and the focus will be on the discrete-time formulation of the problem. Thus, difference equations are used to model the evolution of the system with time, and measurements are assumed to be available at discrete times. For dynamic state estimation, the discrete-time approach is widespread and convenient.

The state space approach to time-series modelling focuses attention on the state vector of a system. The state vector contains all relevant information required to describe the system under investigation. For example, in tracking problems this information could be related to the kinematic characteristics of the target. Alternatively, in an econometrics problem it could be related to monetary flow, interest rates, inflation etc. The measurement vector represents (noisy) observations that are related to the state vector. The measurement vector is generally (but not necessarily) of lower dimension than the state vector. The state space approach is convenient for handling multivariate data and non-linear/non-Gaussian processes and it provides a significant advantage over traditional time-series techniques for these problems. A full description is provided in [41]. Also, many varied examples illustrating the application of non-linear/non-Gaussian state space models are given in [26].

In order to analyse and make inference about a dynamic system at least two models are required. Firstly, a model describing the evolution of the state with time (the system model), and secondly a model relating the noisy measurements to the state (the measurement model). We shall assume that these models are available in a probabilistic form. The probabilistic state space formulation and the requirement for the updating of information on receipt of new measurements are ideally suited for the Bayesian approach. This provides a rigorous general framework for dynamic state estimation problems.

In the Bayesian approach to dynamic state estimation one attempts to construct the posterior probability density function (pdf) of the state based on all available information, including the set of received measurements. Since this pdf embodies all available statistical information, it may be said to be the complete solution to the estimation problem. In principle, an optimal (with respect to any criterion) estimate of the state may be obtained from the pdf. A measure of the accuracy of the estimate may also be obtained. For many problems an estimate is required every time that a measurement is received. In this case a recursive filter is a convenient solution. A recursive filtering approach means that received data can be processed sequentially rather than as a batch, so that it is not neces-

S. Arulampalam is with DSTO, Adelaide, Australia. Email sanjeev.arulampalam@dsto.defence.gov.au

S. Maskell is with the Pattern and Information Processing group, QinetiQ Ltd, Malvern, UK, and Cambridge University Engineering Department, Cambridge, UK. Email: s.maskell@signal.qinetiq.com

N. Gordon is with the Pattern and Information Processing group, QinetiQ Ltd, Malvern, UK. Email: n.gordon@signal.qinetiq.com

T. Clapp is with Astrium, UK. Email: t.clapp@iee.org

©QinetiQ Ltd 2001.©DSTO 2001.©IEEE 2001.
Published with the permission of QinetiQ Ltd, DSTO.

sary to store the complete data set nor to reprocess existing data if a new measurement becomes available[1]. Such a filter consists of essentially two stages: prediction and update. The prediction stage uses the system model to predict the state pdf forwards from one measurement time to the next. Since the state is usually subject to unknown disturbances (modelled as random noise) prediction generally translates, deforms and spreads the state pdf. The update operation uses the latest measurement to modify the prediction pdf. This is achieved using Bayes theorem which is the mechanism for updating knowledge about the target state in the light of extra information from new data.

We begin in Section II with a description of the non-linear tracking problem and its optimal Bayesian solution. When certain constraints hold, this optimal solution is tractable. The Kalman filter and grid-based filter, described in Section III, are two such solutions. Often, the optimal solution is intractable. The methods outlined in Section IV take several different approximation strategies to the optimal solution. These approaches include the extended Kalman filter, approximate grid-based filters and particle filters. Finally, in Section VI, we use a simple scalar example to illustrate some points about the approaches discussed up to this point and then draw conclusions in Section VII. This paper is a tutorial and so, to facilitate easy implementation, 'pseudo-code' for algorithms has been included at relevant points.

II. Non-linear 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 non-linear 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

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

where $\mathbf{h}_k : \Re^{n_x} \times \Re^{n_n} \to \Re^{n_z}$ is a possibly non-linear 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 $\mathbf{z}_{1:k}$ up to time 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:

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

Note that in (3), use has been made of the fact that $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$ as (1) describes a Markov process of order one. The probabilistic model of the state evolution, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, is defined by the system equation (1) and the known statistics of $\mathbf{v}_{k-1}$.

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 $p(\mathbf{z}_k|\mathbf{x}_k)$, defined by the measurement model (2) and the known statistics of $\mathbf{n}_k$. 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[2]. 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 and grid-based filters described in the next section. We also describe how, when the analytic solution

[1] In this paper we assume no out-of-sequence measurements - in the presence of out-of-sequence measurements, the order of times to which the mesaurements relate can differ from the order in which the measurements are processed. For a particle filter solution to the problem of relaxing this assumption see [32].

[2] For clarity, the optimal Bayesian solution solves the problem of recursively calculating the exact posterior density. An optimal algorithm is a method for deducing this solution.

is intractable, extended Kalman filters, approximate grid-based filters, and particle filters, approximate the optimal Bayesian solution.

III. Optimal Algorithms

A. The Kalman Filter

The Kalman filter assumes that the posterior density at every time step is Gaussian and hence parameterised by a mean and covariance.

If $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ is Gaussian, it can be proved that $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is also Gaussian, provided that certain assumptions hold [21]:

- $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are drawn from Gaussian distributions of known parameters
- $\mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1})$ is known and is a linear function of $\mathbf{x}_{k-1}$ and $\mathbf{v}_{k-1}$
- $\mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k)$ is a known linear function of $\mathbf{x}_k$ and $\mathbf{n}_k$

That is, (1) and (2) can be re-written as:

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{6}$$

$$\mathbf{z}_k = H_k \mathbf{x}_k + \mathbf{n}_k \tag{7}$$

F_k and H_k are known matrices defining the linear functions. The covariances of $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are respectively Q_{k-1} and R_k. Here we consider the case when $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ have zero mean and are statistically independent. Note that the system and measurement matrices F_k and H_k, as well as noise parameters Q_{k-1} and R_k, are allowed to be time variant.

The Kalman filter algorithm, derived using (3) and (4), can then be viewed as the following recursive relationship:

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; m_{k-1|k-1}, P_{k-1|k-1}) \tag{8}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; m_{k|k-1}, P_{k|k-1}) \tag{9}$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) \tag{10}$$

where

$$m_{k|k-1} = F_k m_{k-1|k-1} \tag{11}$$

$$P_{k|k-1} = Q_{k-1} + F_k P_{k-1|k-1} F_k^T \tag{12}$$

$$m_{k|k} = m_{k|k-1} + K_k(\mathbf{z}_k - H_k m_{k|k-1}) \tag{13}$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1} \tag{14}$$

and where $\mathcal{N}(x; m, P)$ is a Gaussian density with argument x, mean m and covariance P and:

$$S_k = H_k P_{k|k-1} H_k^T + R_k, \tag{15}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1}, \tag{16}$$

are the covariance of the innovation term $\mathbf{z}_k - H_k m_{k|k-1}$, and the Kalman gain, respectively. In the above equations, the transpose of a matrix M is denoted by M^T.

This is the optimal solution to the tracking problem – if the (highly restrictive) assumptions hold. The implication is that no algorithm can ever do better than a Kalman filter in this linear Gaussian environment. It should be noted that it is possible to derive the same results using a Least Squares (LS) argument [22]. All the distributions are then described by their means and covariances and the algorithm remains unaltered, but are not constrained to be Gaussian. Assuming the means and covariances to be unbiased and consistent, the filter then optimally derives the mean and covariance of the posterior. However, this posterior is not necessarily Gaussian and so, if optimality is the ability of an algorithm to calculate the posterior, the filter is then not certain to be optimal.

Similarly, if smoothed estimates of the states are required, that is estimates of $p(\mathbf{x}_k|\mathbf{z}_{1:k+\ell})$ where $\ell \geq 0$[3], then the Kalman smoother is the optimal estimator of $p(\mathbf{x}_k|\mathbf{z}_{1:k+\ell})$. This holds if ℓ is fixed, *fixed-lag smoothing*, if a batch of data are considered and $0 \leq \ell \leq k$, *fixed-interval smoothing* or if the state at a particular time is of interest, k is fixed, *fixed-point smoothing*. The problem of calculating smoothed densities is of interest because the densities at time k are then conditional not only on measurements up to and including time index k, but also on future measurements. Since there is more information on which to base the estimation, these smoothed densities are typically tighter than the filtered densities.

Though this is true, there is an algorithmic issue that should be highlighted here. It is possible to formulate a backwards-time Kalman filter that recurses through the data sequence from the final data to the first and then combines the estimates from the forward and backward passes to obtain overall smoothed estimates [20]. A different formulation implicitly calculates the backwards-time state estimates and covariances, recursively estimating the smoothed quantities [38]. Both techniques are prone to having to calculate matrix inverses that do not necessarily exist. Instead, it is preferable to propagate different quantities using an information filter when carrying out the backwards-time recursion [4].

[3] If $\ell = 0$ then the problem reduces to the estimation of $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ considered up to this point.

B. Grid-based methods

Grid-based methods provide the optimal recursion of the filtered density, $p(\mathbf{x}_k|\mathbf{z}_{1:k})$, if the state space is discrete and consists of a finite number of states. Suppose the state space at time $k-1$ consists of discrete states $\mathbf{x}_{k-1}^i$, $i = 1, \ldots, N_s$. For each state $\mathbf{x}_{k-1}^i$, let the conditional probability of that state, given measurements up to time $k-1$ be denoted by $w_{k-1|k-1}^i$, that is, $Pr(\mathbf{x}_{k-1} = \mathbf{x}_{k-1}^i|\mathbf{z}_{1:k-1}) = w_{k-1|k-1}^i$. Then, the posterior pdf at $k-1$ can be written as

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) = \sum_{i=1}^{N_s} w_{k-1|k-1}^i \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^i) \quad (17)$$

where $\delta(\cdot)$ is the Dirac delta measure. Substitution of (17) into (3) and (4) yields the prediction and update equations, respectively:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \sum_{i=1}^{N_s} w_{k|k-1}^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (18)$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \sum_{i=1}^{N_s} w_{k|k}^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (19)$$

where

$$w_{k|k-1}^i \triangleq \sum_{j=1}^{N_s} w_{k-1|k-1}^j p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^j), \quad (20)$$

$$w_{k|k}^i \triangleq \frac{w_{k|k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i)}{\sum_{j=1}^{N_s} w_{k|k-1}^j p(\mathbf{z}_k|\mathbf{x}_k^j)} \quad (21)$$

The above assumes that $p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^j)$ and $p(\mathbf{z}_k|\mathbf{x}_k^i)$ are known, but does not constrain the particular form of these discrete densities. Again, this is the optimal solution if the assumptions made hold.

IV. Sub-optimal Algorithms

In many situations of interest, the assumptions made above do not hold. The Kalman filter and grid-based methods cannot therefore be used as described – approximations are necessary. In this section, we consider three approximate non-linear Bayesian filters: a) extended Kalman filter, b) Approximate grid-based methods, and c) Particle filters.

A. Extended Kalman Filter

If (1) and (2) cannot be rewritten in the form of (6) and (7) because the functions are non-linear, then a local linearisation of the equations may be a sufficient description of the non-linearity. The extended Kalman filter (EKF), is based upon this approximation. $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ is approximated by a Gaussian:

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \approx \mathcal{N}(\mathbf{x}_{k-1}; m_{k-1|k-1}, P_{k-1|k-1}) \quad (22)$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) \approx \mathcal{N}(\mathbf{x}_k; m_{k|k-1}, P_{k|k-1}) \quad (23)$$

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \mathcal{N}(\mathbf{x}_k; m_{k|k}, P_{k|k}) \quad (24)$$

where

$$m_{k|k-1} = \mathbf{f}_k(m_{k-1|k-1}) \quad (25)$$

$$P_{k|k-1} = Q_{k-1} + \hat{F}_k P_{k-1|k-1} \hat{F}_k^T \quad (26)$$

$$m_{k|k} = m_{k|k-1} + K_k(\mathbf{z}_k - \mathbf{h}_k(m_{k|k-1})) \quad (27)$$

$$P_{k|k} = P_{k|k-1} - K_k \hat{H}_k P_{k|k-1} \quad (28)$$

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

$$\hat{F}_k = \left.\frac{d\mathbf{f}_k(x)}{dx}\right|_{x=m_{k-1|k-1}} \quad (29)$$

$$\hat{H}_k = \left.\frac{d\mathbf{h}_k(x)}{dx}\right|_{x=m_{k|k-1}} \quad (30)$$

$$S_k = \hat{H}_k P_{k|k-1} \hat{H}_k^T + R_k \quad (31)$$

$$K_k = P_{k|k-1} \hat{H}_k^T S_k^{-1} \quad (32)$$

The EKF as described above utilises the first term in a Taylor expansion of the non-linear 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 [23], [42], [43]. The resulting filter, known as the "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 non-linearity 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 non-linearity; 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 [1].

B. Approximate Grid-Based Methods

If the state space is continuous, but can be decomposed into N_s 'cells', $\{\mathbf{x}_k^i : i = 1, \ldots, N_s\}$, then a grid-based method can be used to approximate the posterior density. Specifically, suppose the approximation to the posterior pdf at $k-1$ is given by

$$p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1}) \approx \sum_{i=1}^{N_s} w_{k-1|k-1}^i \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^i) \tag{33}$$

Then, the prediction and update equations can be written as

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

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_{k|k}^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{35}$$

where

$$w_{k|k-1}^i \triangleq \sum_{j=1}^{N_s} w_{k-1|k-1}^j \int_{\mathbf{x}\in\mathbf{x}_k^i} p(\mathbf{x}|\bar{\mathbf{x}}_{k-1}^j) d\mathbf{x}, \tag{36}$$

$$w_{k|k}^i \triangleq \frac{w_{k|k-1}^i \int_{\mathbf{x}\in\mathbf{x}_k^i} p(\mathbf{z}_k|\mathbf{x}) d\mathbf{x}}{\sum_{j=1}^{N_s} w_{k|k-1}^j \int_{\mathbf{x}\in\mathbf{x}_k^j} p(\mathbf{z}_k|\mathbf{x}) d\mathbf{x}} \tag{37}$$

Here $\bar{\mathbf{x}}_{k-1}^j$ denotes the centre of the j-th cell at time index $k-1$. The integrals in (36) and (37) arise due to the fact that the grid points $\mathbf{x}_k^i$, $i = 1, \ldots, N_s$, represent regions of continuous state space, and thus the probabilities must be integrated over these regions. In practice, to simplify computation, a further approximation is made in the evaluation of $w_{k|k}^i$. Specifically, these weights are computed at the centre of the 'cell' corresponding to $\mathbf{x}_k^i$:

$$w_{k|k-1}^i \triangleq \sum_{j=1}^{N_s} w_{k-1|k-1}^j p(\bar{\mathbf{x}}_k^i|\bar{\mathbf{x}}_{k-1}^j), \tag{38}$$

$$w_{k|k}^i \approx \frac{w_{k|k-1}^i p(\mathbf{z}_k|\bar{\mathbf{x}}_k^i)}{\sum_{j=1}^{N_s} w_{k|k-1}^j p(\mathbf{z}_k|\bar{\mathbf{x}}_k^j)} \tag{39}$$

The grid must be sufficiently dense to get a good approximation to the continuous state space. As the dimensionality of the state space increases, the computational cost of the approach therefore increases dramatically. If the state space is not finite in extent, then using a grid-based approach necessitates some truncation of the state space. Another disadvantage of grid-based methods is that the state space must be predefined and therefore cannot be partitioned unevenly to give greater resolution in high probability density regions, unless prior knowledge is used.

Hidden Markov model (HMM) filters [30], [35], [36], [39] are an application of such approximate grid-based methods in a fixed-interval smoothing context and have been used extensively in speech processing. In HMM based tracking, a common approach is to use the Viterbi algorithm [18] to calculate the maximum a-posteriori estimate of the path through the trellis, that is the sequence of discrete states that maximises the probability of the state sequence given the data. Another approach, due to Baum-Welch [35], is to calculate the probability of each discrete state at each time epoch given the entire data sequence[4].

V. Particle Filtering Methods

A. 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 [29], particle filtering [6], interacting particle approximations [10], [11] and survival of the fittest [24]. 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

$$p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i), \tag{40}$$

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

[4]The Viterbi and Baum-Welch algorithms are frequently applied when the state space is approximated to be discrete. The algorithms are optimal if and only if the underlying state space is truly discrete in nature.

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

$$p(x) \approx \sum_{i=1}^{N_s} w^i \delta(x - x^i) \tag{41}$$

where

$$w^i \propto \frac{\pi(x^i)}{q(x^i)} \tag{42}$$

is the normalised weight of the i-th particle.

So, if the samples, $\mathbf{x}_{0:k}^i$, were drawn from an importance density, $q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ then the weights in (40) are defined by (42) to be

$$w_k^i \propto \frac{p(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^i|\mathbf{z}_{1:k})} \tag{43}$$

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

$$q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \tag{44}$$

then one can obtain samples $\mathbf{x}_{0:k}^i \sim q(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$ by augmenting each of the existing samples $\mathbf{x}_{0:k-1}^i \sim q(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1})$ with the new state $\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$. 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 $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Note that (4) can be derived by integrating (45).

$$\begin{aligned} p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}) =& \frac{p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \\ =& \frac{p(\mathbf{z}_k|\mathbf{x}_{0:k}, \mathbf{z}_{1:k-1}) p(\mathbf{x}_k|\mathbf{x}_{0:k-1}, \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} \\ & \times p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \qquad (45) \\ =& \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})} p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \\ \propto& p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{0:k-1}|\mathbf{z}_{1:k-1}) \qquad (46) \end{aligned}$$

By substituting (44) and (46) into (43), the weight update equation can then be shown to be

$$\begin{aligned} w_k^i \propto& \frac{p(\mathbf{z}_k|\mathbf{x}_k^i) p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i) p(\mathbf{x}_{0:k-1}^i|\mathbf{z}_{1:k-1})}{q(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1}^i|\mathbf{z}_{1:k-1})} \\ =& w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i) p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k})} \end{aligned} \tag{47}$$

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 $\mathbf{x}_{k-1}$ 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 $\mathbf{x}_k^i$ need be stored, and so one can discard the path, $\mathbf{x}_{0:k-1}^i$, and history of observations, $\mathbf{z}_{1:k-1}$. The modified weight is then

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i) p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)} \tag{48}$$

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

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{49}$$

where the weights are defined in (48). It can be shown that as $N_s \to \infty$ the approximation (49) 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 (48)
- END FOR

A.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 [28], and defined as

$$N_{eff} = \frac{N_s}{1 + \text{Var}(w_k^{*i})}, \tag{50}$$

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

$$\widehat{N_{eff}} = \frac{1}{\sum_{i=1}^{N_s}(w_k^i)^2} \tag{51}$$

where w_k^i is the normalised weight obtained using (47). 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.

A.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 $\mathrm{Var}(w_k^{*i})$ 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

$$\begin{aligned} q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)_{opt} =& p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k) \\ =& \frac{p\left(\mathbf{z}_k|\mathbf{x}_k, \mathbf{x}_{k-1}^i\right) p\left(\mathbf{x}_k|\mathbf{x}_{k-1}^i\right)}{p\left(\mathbf{z}_k|\mathbf{x}_{k-1}^i\right)}. \end{aligned} \tag{52}$$

Substitution of (52) into (48) yields

$$\begin{aligned} w_k^i &\propto w_{k-1}^i p\left(\mathbf{z}_k|\mathbf{x}_{k-1}^i\right) \\ &= w_{k-1}^i \int p\left(\mathbf{z}_k|\mathbf{x}_k'\right) p\left(\mathbf{x}_k'|\mathbf{x}_{k-1}^i\right) d\mathbf{x}_k'. \end{aligned} \tag{53}$$

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 (53) 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], whereby the discrete modal state (defining the maneuver index) is tracked using a particle filter and (conditioned on the maneuver index) the continuous base state is tracked using a Kalman filter.

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 non-linear and the measurements linear. Such a system is given by

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1}, \tag{54}$$

$$\mathbf{z}_k = H_k\mathbf{x}_k + \mathbf{n}_k, \tag{55}$$

where

$$\mathbf{v}_{k-1} \sim \mathcal{N}(\mathbf{v}_{k-1}; \mathbf{0}_{n_v\times 1}, Q_{k-1}), \tag{56}$$

$$\mathbf{n}_k \sim \mathcal{N}(\mathbf{n}_k; \mathbf{0}_{n_n\times 1}, R_k), \tag{57}$$

and $\mathbf{f}_k : \Re^{n_x} \to \Re^{n_x}$ is a non-linear 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

$$\mathbf{\Sigma}_k^{-1} = Q_{k-1}^{-1} + H_k^T R_k^{-1} H_k \tag{58}$$

$$\mathbf{m}_k = \mathbf{\Sigma}_k \left(Q_{k-1}^{-1}\mathbf{f}_k(\mathbf{x}_{k-1}) + H_k^T R_k^{-1}\mathbf{z}_k\right), \tag{59}$$

one obtains

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k) = \mathcal{N}(\mathbf{x}_k; \mathbf{m}_k, \mathbf{\Sigma}_k) \tag{60}$$

and

$$p(\mathbf{z}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{z}_k; H_k\mathbf{f}_k(\mathbf{x}_{k-1}), Q_{k-1} + H_k R_k H_k^T). \tag{61}$$

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 [40]. 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.

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}^i) \tag{62}$$

Substitution of (62) into (48) then yields

$$w_k^i \propto w_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i). \tag{63}$$

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, as illustrated by section VI, the choice is the crucial design step in the design of a particle filter.

A.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

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{64}$$

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 (64), 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 [37], [6]. Note that other efficient (in terms of reduced MC variation) resampling schemes such as stratified sampling and residual sampling [28] may be applied as alternatives to this algorithm. Systematic resampling [25] 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 $\mathbb{U}[a,b]$ is the Uniform distribution on the interval $[a,b]$ (inclusive of the limits). 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 V-B.2.

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

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[5]. Thirdly, since the diversity of the paths of the particles is reduced, any smoothed estimates based on the particles' paths degenerate[6]. 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].

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: $u_1 \sim \mathbb{U}\left[0, N_s^{-1}\right]$
- 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 (48)
- 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 (51)
- 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

There have been some systematic techniques proposed

[5]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], [27].

[6]Since the particles actually represent paths through the state space, by storing the trajectory taken by each particle, fixed-lag and fixed-point smoothed estimates of the state can be obtained [4].

recently to solve the problem of sample impoverishment. One such technique is the *resample-move* algorithm [19], which is not be described in detail in this paper. 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 asymtotically approximate samples from the posterior and so is the method of choice of the authors. An alternative solution to the same problem is *regularisation* [31], which is discussed in Section V-B.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.

A.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 [33] 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 [29] 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.

B. 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).

B.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, and 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: $q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_{1:k})$ is chosen to be the prior density $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$, and (ii) Resampling step: to be applied at every time index.

The above choice of importance density implies that we need samples from $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$. A sample $\mathbf{x}_k^i \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ can be generated by first generating a process noise sample $\mathbf{v}_{k-1}^i \sim p_v(\mathbf{v}_{k-1})$ and setting $\mathbf{x}_k^i = \mathbf{f}_k(\mathbf{x}_{k-1}^i, \mathbf{v}_{k-1}^i)$, 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

$$w_k^i \propto w_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i). \tag{65}$$

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

$$w_k^i \propto p(\mathbf{z}_k|\mathbf{x}_k^i). \tag{66}$$

The weights given by the proportionality in (66) are normalised before the resampling stage. An iteration of the algorithm is then described by algorithm 4.

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.

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

B.2 Auxiliary Sampling Importance Resampling filter

The Auxiliary Sampling Importance Resampling (ASIR) filter was introduced by Pitt & Shephard [34] 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

$$\begin{aligned} p(\mathbf{x}_k, i|\mathbf{z}_{1:k}) &\propto p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k, i|\mathbf{z}_{1:k-1}) \\ &= p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|i, \mathbf{z}_{1:k-1}) p(i|\mathbf{z}_{1:k-1}) \\ &= p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{x}_{k-1}^i) w_{k-1}^i \end{aligned} \tag{67}$$

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

$$q(\mathbf{x}_k, i|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mu_k^i) p(\mathbf{x}_k|\mathbf{x}_{k-1}^i) w_{k-1}^i \tag{68}$$

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

$$q(\mathbf{x}_k, i|\mathbf{z}_{1:k}) = q(i|\mathbf{z}_{1:k}) q(\mathbf{x}_k|i, \mathbf{z}_{1:k}), \tag{69}$$

and defining

$$q(\mathbf{x}_k|i, \mathbf{z}_{1:k}) \triangleq p(\mathbf{x}_k|\mathbf{x}_{k-1}^i), \tag{70}$$

it follows from (68) that

$$q(i|\mathbf{z}_{1:k}) \propto p(\mathbf{z}_k|\mu_k^i) w_{k-1}^i. \tag{71}$$

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 (67) to (68):

$$\begin{aligned} w_k^j &\propto w_{k-1}^{i^j} \frac{p(\mathbf{z}_k|\mathbf{x}_k^j) p(\mathbf{x}_k^j|\mathbf{x}_{k-1}^{i^j})}{q(\mathbf{x}_k^j, i^j|\mathbf{z}_{1:k})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k^j)}{p(\mathbf{z}_k|\mu_k^{i^j})} \end{aligned} \tag{72}$$

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 (72)
- 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 [34] 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 ap-

proximation of $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$. In such scenarios, the use of ASIR then degrades performance.

B.3 The Regularised Particle Filter

Recall that resampling was suggested in section V-B.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", a sever case of sample impoverishment 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 [31] 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 (64). Specifically, in the RPF, samples are drawn from the approximation

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i K_h(\mathbf{x}_k - \mathbf{x}_k^i) \tag{73}$$

where

$$K_h(\mathbf{x}) = \frac{1}{h^{n_x}} K\left(\frac{\mathbf{x}}{h}\right) \tag{74}$$

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 (73), defined as

$$\text{MISE}(\hat{p}) = \mathbb{E}\left[\int \left[\hat{p}(\mathbf{x}_k|\mathbf{z}_{1:k}) - p(\mathbf{x}_k|\mathbf{z}_{1:k})\right]^2 d\mathbf{x}_k\right] \tag{75}$$

where $\hat{p}(\cdot|\cdot)$ denotes the approximation to $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ given by the RHS of (73)[7]. In the special case of all the samples having the same weight, the optimal choice of the Kernel is the Epanechnikov Kernel [31],

$$K_{opt} = \begin{cases} \frac{n_x+2}{2c_{n_x}}(1 - \|\mathbf{x}\|^2) & \text{if } \|\mathbf{x}\| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{76}$$

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 [31]

$$h_{opt} = A N_s^{\frac{1}{n_x+4}} \tag{77}$$

$$A = [8c_{n_x}^{-1}(n_x+4)(2\sqrt{\pi})^{n_x}]^{\frac{1}{n_x+4}} \tag{78}$$

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 (48)
- 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 (51)
- 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

Though the results of (76) and (77)-(78) 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

[7] As observed by one of the anonymous reviewers, it is worth noting that the use of the Kernel approximation become increasingly less appropriate as the dimensionality of the state increases.

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.

By following the above procedure, we generate an i.i.d random sample $\{\mathbf{x}_k^{i\,*}\}_{i=1}^{N_s}$ drawn from (73).

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 asymtotically 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.

VI. Example

Here we consider the following set of equations as an illustrative example:

$$p\left(\mathbf{x}_k|\mathbf{x}_{k-1}\right) = \mathcal{N}\left(\mathbf{x}_k; \mathbf{f}_k\left(\mathbf{x}_{k-1}, k\right), Q_{k-1}\right) \tag{79}$$

$$p\left(\mathbf{z}_k|\mathbf{x}_k\right) = \mathcal{N}\left(\mathbf{z}_k; \frac{\mathbf{x}_k^2}{20}, R_k\right) \tag{80}$$

or equivalently

$$\mathbf{x}_k = \mathbf{f}_k\left(\mathbf{x}_{k-1}, k\right) + \mathbf{v}_{k-1} \tag{81}$$

$$\mathbf{z}_k = \frac{\mathbf{x}_k^2}{20} + \mathbf{n}_k \tag{82}$$

where

$$\mathbf{f}_k\left(\mathbf{x}_{k-1}, k\right) = \frac{\mathbf{x}_{k-1}}{2} + \frac{25\mathbf{x}_{k-1}}{1+\mathbf{x}_{k-1}^2} + 8\cos(1.2k) \tag{83}$$

and where $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are zero mean Gaussian random variables with variances Q_{k-1} and R_k respectively. We use $Q_{k-1} = 10$ and $R_k = 1$. This example has been analysed before in many publications [5], [17], [25].

We consider the performance of the algorithms detailed in table I. In order to qualitatively gauge performance and discuss resulting issues, we consider one exemplar run. In order to quantify performance, we use the traditional measure of performance, the Root Mean Squared Error, RMSE. It should be noted that this measure of performance is not exceptionally meaningful for this multimodal problem. However, it has been used extensively in the literature and is included here for that reason and because it facilitates quantitative comparison.

For reference, the true states for the exemplar run are shown in figure 1 and the measurements in figure 2.

The approximate grid-based method uses 50 states with centres equally spaced on $[-25, 25]$. All the particle filters have 50 particles and employ resampling at every time step ($N_T = N_s$). The auxiliary particle filter uses $\mu_k^i \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$. The regularised particle filter uses the Kernel and bandwidth described in section V-B.3.

To visualise the densities inferred by the approximate grid-based and particle filters, the total probability mass at any time in each of 50 equally spaced regions on $[-25, 25]$ is shown as images in figures 5 through 9. At any given time (and so in any vertical slice through the image), darker regions represent higher probability than lighter regions. A graduated scale relating intensity to probability mass in a pixel is shown next to each image.

A. Extended Kalman Filter

The extended Kalman filter's local linearisation and Gaussian approximation are not a sufficient description of the non-linear and non-Gaussian nature of the example. Once the EKF cannot adequately approximate the bimodal nature of the underlying posterior, the Gaussian approximation fails - the EKF is prone to either choosing the 'wrong' mode or just sitting on the average between the modes. As a result of this inability to adequately approximate the density, the linearisation approximation becomes poor.

This can be seen from figure 3. The mean of the filter is rarely close to the true state. Were the density to be Gaussian, one would expect the state to be within two standard deviations of the mean approximately 95% of the time. From figure 4, it is evident that there are times when the distribution is sufficiently broad to capture the true state in this region, but that there are also times when the filter becomes highly over-confident of a biased estimate of the state. The implication of this is that it is very difficult to detect inconsistent EKF errors automatically on-line.

The RMSE measure indicates that the EKF is the least accurate of the algorithms at approximating the posterior. The approximations made by the EKF are inappropriate in this example.

B. Approximate Grid-Based Filter

This example is low dimensional and so one would expect that an approximate grid-based approach would perform well. Figure 5 shows this is indeed the case. The grid-based approximation is able to model the multi-modality of the problem.

Using the approximate grid-based filter rather than an EKF yields a marked reduction in RMS errors. A particle filter with N_s particles conducts $O(N_s)$ operations per iteration whereas an approximate grid-based filter carries out $O(N_s^2)$ operations with N_s cells. It is therefore surprising

Algorithm	Proposal	Section	Figures	RMSE
Extended Kalman filter	N/A	IV-A	3,4	23.19
Approximate grid-Based Filter	N/A	IV-B	5	6.09
SIR Particle filter	$p(\mathbf{x}_k\|\mathbf{x}_{k-1}^i)$	V-B.1	6	5.54
Auxiliary Particle filter	$p(\mathbf{x}_k\|\mathbf{x}_{k-1}^i)$	V-B.2	7	5.35
Regularised Particle filter	$p(\mathbf{x}_k\|\mathbf{x}_{k-1}^i)$	V-B.3	8	5.55
'Likelihood' Particle filter	$p(\mathbf{x}_k\|\mathbf{s}_k)p(\mathbf{s}_k\|\mathbf{z}_k)$	A	9	5.30

TABLE I

TABLE OF THE ALGORITHMS USED, THE SECTIONS OF THE ARTICLE AND FIGURES WHICH RELATE TO THE ALGORITHMS, AND RMSE VALUES (AVERAGED OVER 100 MC RUNS).

that the RMS errors for the approximate grid are larger than those of the particle filter. The authors suspect this is an artifact of the grid being fixed - the resolution of the algorithm is predefined and the fixed position of the grid points means that the grid points near ± 25 contribute significantly to the error when the true state is far from these values.

C. SIR Particle Filter

Using the prior distribution as the importance density is in some sense regarded as a standard SIR particle filter and so is an appropriate particle filter algorithm with which to begin. As can be seen from figure 6, the SIR particle filter gives disappointing results with the low number of particles used here. The speckled appearance of the figure is a result of sampling a low number of particles from the (broad) prior. It is an artifact resulting from the inadequate amount of sampling.

The RMSE metric shows a marginal improvement over the approximate grid-based filter. To achieve smaller errors one could simply increase the number of particles, but here we shall now investigate the effect of using the alternative particle filter algorithms described up to this point.

D. Auxiliary Particle Filter

One way to reduce errors might be that the proposed particle positions are chosen badly. One might therefore think that choosing the proposed particles in a more intelligent manner would yield better results. An auxiliary particle filter would then seem to be an appropriate candidate replacement algorithm for SIR. Here we have μ_k^i as a sample from $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$.

As shown by figure 7, for this example, the auxiliary particle filter performs well. There is arguably less speckle in figure 7 than in 6 and the probability mass appears to be better concentrated around the true state. However, one might think this problem is not very well suited to an auxiliary particle filter since the prior is often much broader than the likelihood. When the prior is broad, those particles with a noise realisation that happens to have a high likelihood are resampled many times. There is no guarantee that other samples from the prior will also lie in the same region of the state space since only a single point is being used to characterise the filtered density for each particle.

The RMS errors are slightly reduced from those for SIR.

E. Regularised Particle Filter

Using the regularised particle filter results in a smoothing of the approximation to the posterior. This is apparent from figure 8. The speckle is reduced and the peaks broadened when compared to the previous particle filters' images.

The regularised particle filter gives very similar RMS errors to the SIR particle filter. The regularisation does not result in a significant reduction in errors for this data set.

F. 'Likelihood' Particle Filter

All the aforementioned particle filters share the prior as a proposal density. For this example, much of the time, the likelihood is far tighter than the prior. As a result, the posterior is closer in similarity to the likelihood than to the prior. The importance density is an approximation to the posterior. So, using a better approximation based on the likelihood, rather than the prior, can be expected to improve performance.

Figure 9 shows that the use of such an importance density (see Appendix for details) yields a reduction in speckle and that the peaks of the density are closer on average to the true state than for any of the other particle filters.

The RMS errors are similar to those for the Auxiliary particle filter.

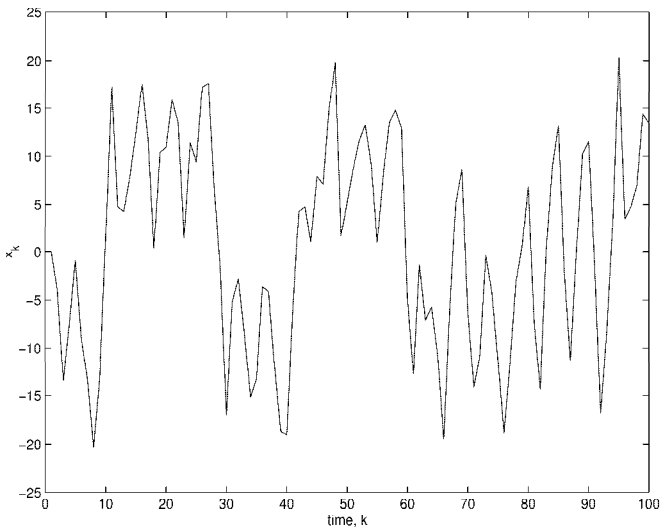

Fig. 1. Figure of the true values of the state, $\mathbf{x}_k$, as a function of k for the exemplar run.

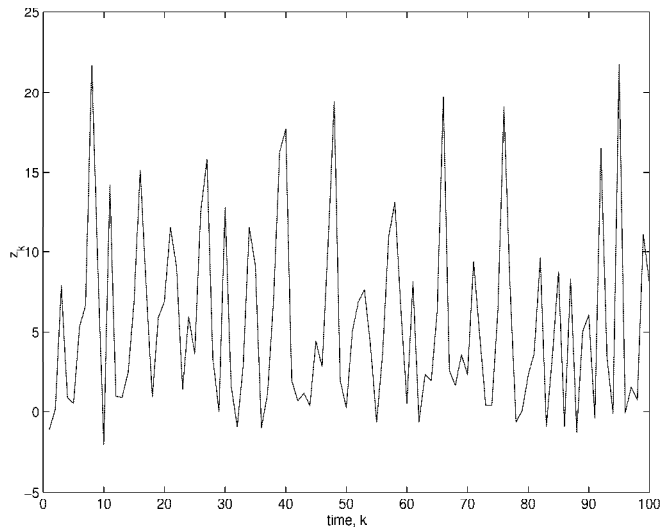

Fig. 2. Figure of the measurements, $\mathbf{z}_k$, of the states, $\mathbf{x}_k$, shown in figure 1 for the same exemplar run.

G. Crucial Step in the Application of a Particle Filter

The RMS errors indicate that, in highly non-linear environments, a non-linear filter such as an approximate grid-based filter or particle filter offers an improvement in performance over an EKF. This improvement results from approximating the density rather than the models.

When using a particle filter, one can often expect and frequently achieve an improvement in performance by using far more particles or alternatively by employing regularisation or using an auxiliary particle filter. For this example, a slight improvement in RMS errors is possible by using an importance density other than $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$. The authors assert that an importance density tuned to a particular problem will yield an appropriate trade off between the number of particles and the computational expense necessary for each particle, giving the best qualitative performance with affordable computational effort.

The crucial point to convey is that all the refinements of the particle filter assume that the choice of importance density has already been made. Choosing the importance density to be well suited to a given application requires careful thought. The choice made is crucial.

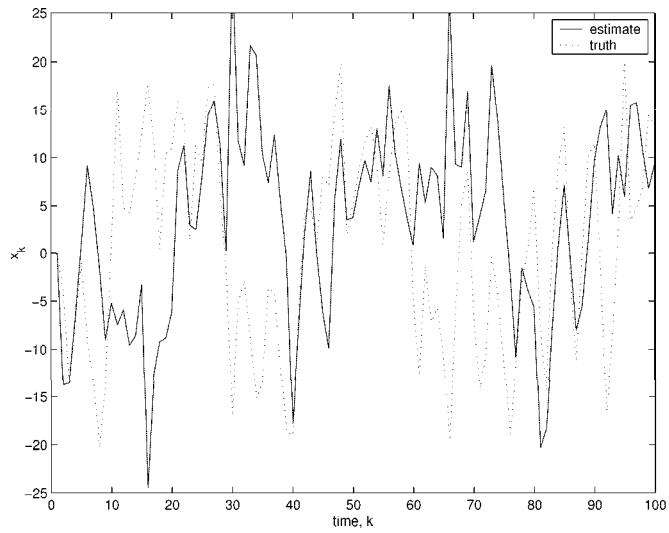

Fig. 3. Figure of the evolution of the EKF's mean estimate of the state.

VII. Conclusions

For a particular problem, if the assumptions of the Kalman filter or grid-based filters hold then no other algorithm can out-perform them. 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. Approximate grid-based filters approximate the continuous state space as a set of discrete regions. This necessitates the pre-definition of these regions and becomes prohibitively computationally expensive when dealing with high dimensional state spaces [3]. 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.

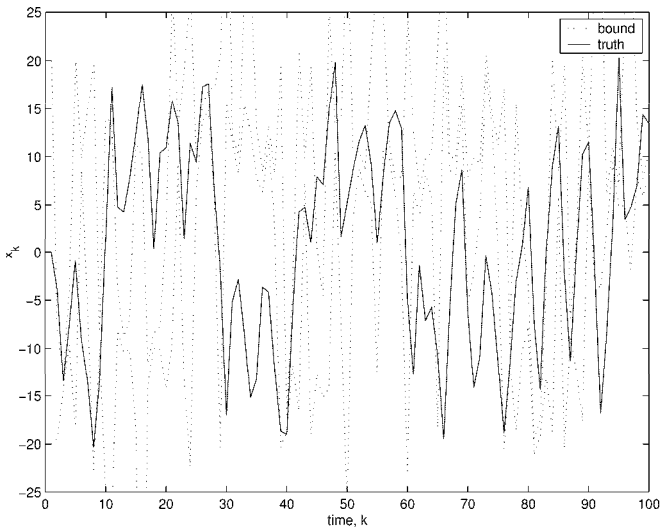

Fig. 4. Figure of the evolution of the upper and lower 2σ positions of the state as estimated by the EKF (dotted) with the true state also shown (solid).

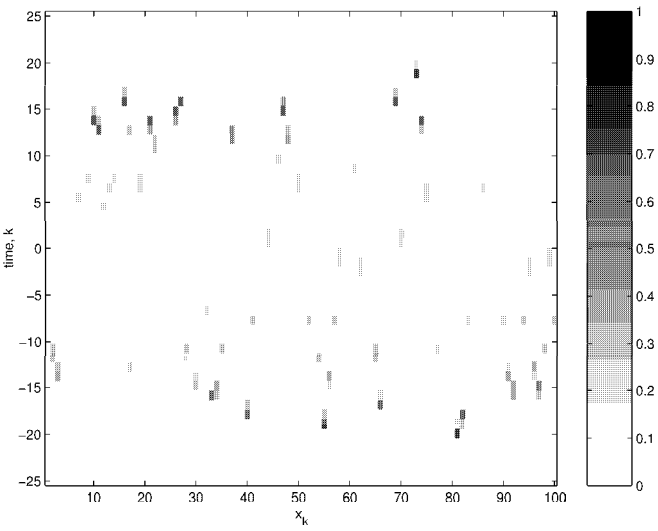

Fig. 5. Image representing evolution of probability density for approximate grid-based filter.

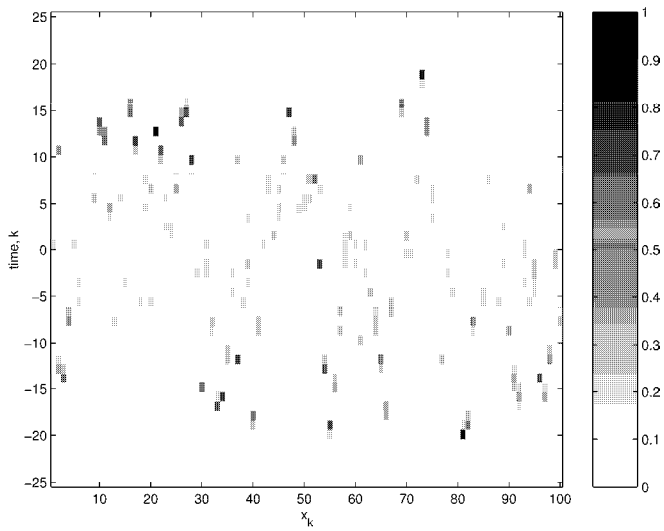

Fig. 6. Image representing evolution of probability density for SIR Particle Filter.

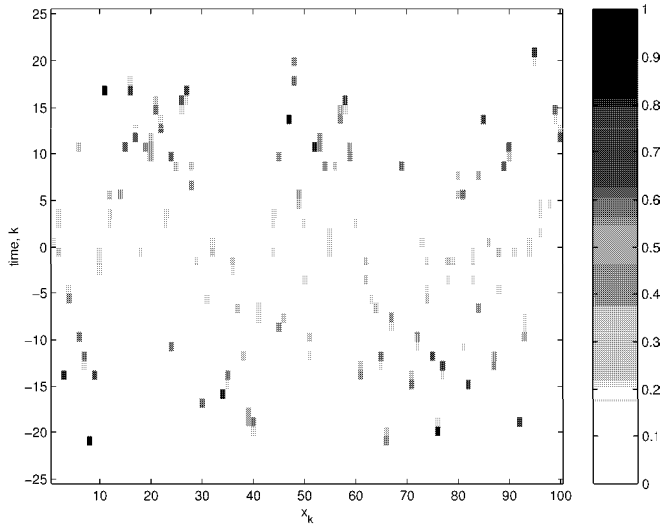

Fig. 7. Image representing evolution of probability density for Auxiliary Particle Filter.

Acknowledgements

The authors are grateful to the anonymous reviewers and the editors of this special issue for their many helpful suggestions which have greatly improved the presentation of this paper. The authors would also like to thank various funding sources who have contributed to this research. The first author is grateful to the Royal Academy of Engineering for an Anglo-Australian post-doctoral research fellowship. The second author thanks the Royal Commission for the Exhibition of 1851 for his Industrial Fellowship. The third author thanks QinetiQ Ltd.

Appendix

I. Importance Density for 'Likelihood' Particle Filter

This appendix describes the importance density for the 'Likelihood' particle filter, which is intended to illustrate the crucial nature of the choice of importance density in a particle filter. This importance density is not intended to be generically applicable, but to be one chosen to work well for the specific problem and parameters described in section VI.

To keep the notation simple, throughout this appendix $\mathbf{s}_k = (\mathbf{x}_k)^2$. For a uniform prior on $\mathbf{s}_k$, the density $p(\mathbf{s}_k|\mathbf{z}_k)$

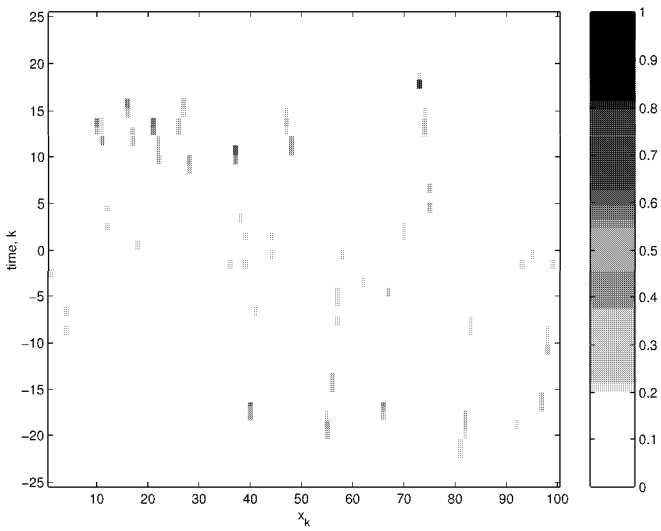

Fig. 8. Image representing evolution of probability density for Regularised Particle Filter.

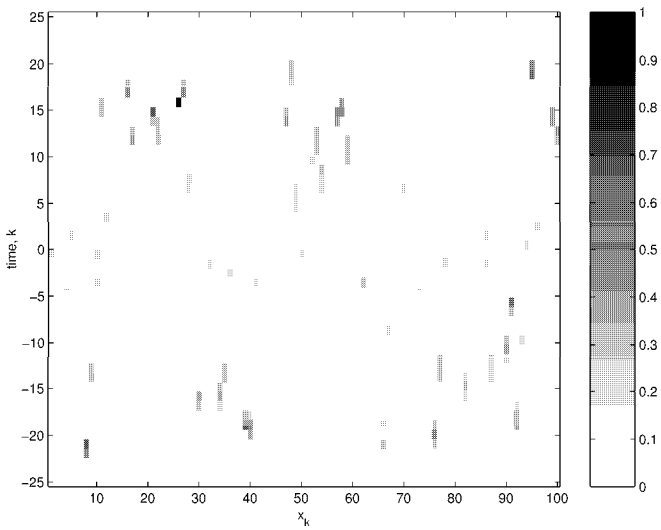

Fig. 9. Image representing evolution of probability density for 'Likelihood' Particle Filter.

can be written by Bayes rule as

$$p\left(\mathbf{s}_k|\mathbf{z}_k\right) \propto \begin{cases} p\left(\mathbf{z}_k|\mathbf{s}_k\right) & \mathbf{s}_k \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{84}$$

We can then sample $\mathbf{s}_k^i \sim p\left(\mathbf{s}_k|\mathbf{z}_k\right)$ (samples, $\mathbf{s}_k^i$, are repeatedly drawn from $\hat{p}\left(\mathbf{s}_k|\mathbf{z}_k\right) \propto p\left(\mathbf{z}_k|\mathbf{s}_k\right)$ until one is drawn such that $p(\mathbf{s}_k|\mathbf{z}_k) > 0$, ie. one such that $\mathbf{s}_k \geq 0$). Then $p(\mathbf{x}_k|\mathbf{s}_k^i)$ can be chosen to be a pair of delta functions.

$$p(\mathbf{x}_k|\mathbf{s}_k^i) = \frac{\delta\left(\mathbf{x}_k - \sqrt{\mathbf{s}_k^i}\right) + \delta\left(\mathbf{x}_k + \sqrt{\mathbf{s}_k^i}\right)}{2} \tag{85}$$

This can then be used to form a 'Likelihood' based importance density which samples $\mathbf{x}_k^i$ conditional on $\mathbf{z}_k$ and independently from $\mathbf{x}_{k-1}^i$.

$$q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_{1:k}) \propto p(\mathbf{x}_k|\mathbf{s}_k)p(\mathbf{s}_k|\mathbf{z}_k) \tag{86}$$

The weight of the sample can be calculated according to (47).

$$w_k^i \propto w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_{1:k})} \tag{87}$$

$$= w_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{p(\mathbf{x}_k^i|\mathbf{s}_k^i)p(\mathbf{s}_k^i|\mathbf{z}_k)} \tag{88}$$

$$= w_{k-1}^i \frac{p(\mathbf{x}_k^i|\mathbf{z}_k)p(\mathbf{z}_k)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{p(\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{s}_k^i)p(\mathbf{s}_k^i|\mathbf{z}_k)} \tag{89}$$

Now, $p(\mathbf{x}_k^i|\mathbf{s}_k^i) = \frac{1}{2}$, $p(\mathbf{z}_k)$ and $p(\mathbf{x}_k^i)$ are constant, so disappear, leaving

$$w_k^i \propto w_{k-1}^i p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)\frac{p(\mathbf{x}_k^i|\mathbf{z}_k)}{p(\mathbf{s}_k^i|\mathbf{z}_k)} \tag{90}$$

Now, the ratio of $p(\mathbf{x}_k^i|\mathbf{z}_k)$ to $p(\mathbf{s}_k^i|\mathbf{z}_k)$ needs careful consideration. Though the values of $p(\mathbf{x}_k^i|\mathbf{z}_k)$ and $p(\mathbf{s}_k^i|\mathbf{z}_k)$ might be initially thought to be proportional, they are probability densities defined with respect to a different measure (ie. a different parameterisation of the space). Since $p(\mathbf{x}_k|\mathbf{z}_k)$ integrates to unity over $d\mathbf{x}_k$ while $p(\mathbf{s}_k|\mathbf{z}_k)$ integrates to unity over $d\mathbf{s}_k$, the ratio of the probability densities is then proportional to the inverse of the ratio of the lengths, $d\mathbf{x}_k$ and $d\mathbf{s}_k$. The ratio of $p(\mathbf{x}_k^i|\mathbf{z}_k)$ to $p(\mathbf{s}_k^i|\mathbf{z}_k)$ is the determinant of the Jacobian of the transformation from $\mathbf{s}_k$ to $\mathbf{x}_k$.

$$\frac{p(\mathbf{x}_k^i|\mathbf{z}_k)}{p(\mathbf{s}_k^i|\mathbf{z}_k)} \propto \left|\frac{d\mathbf{s}_k}{d\mathbf{x}_k}\right| = 2\mathbf{x}_k \tag{91}$$

An expression for the weight is then forthcoming

$$w_k^i \propto w_{k-1}^i p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)\frac{p(\mathbf{x}_k^i|\mathbf{z}_k)}{p(\mathbf{s}_k^i|\mathbf{z}_k)} \propto w_{k-1}^i p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)\mathbf{x}_k^i \tag{92}$$

The particle filter that results from this sampling procedure is given in algorithm 7.

So, rather than draw samples from the state evolution distribution and then weight them according to their likelihood, samples are drawn from the likelihood and then assigned weights on the basis of the state evolution distribution.

ALGORITHM 7: 'LIKELIHOOD' PARTICLE FILTER

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

- FOR $i = 1 : N_s$
  - REPEAT
    - Draw $\mathbf{s}_k^i \sim \hat{p}(\mathbf{s}_k|\mathbf{z}_k) \propto p(\mathbf{z}_k|\mathbf{s}_k)$
  - UNTIL $\mathbf{s}_k^i \geq 0$
  - IF $\mathbb{U}[0,1] > \frac{1}{2}$
    - $\mathbf{x}_k^i = \sqrt{\mathbf{s}_k^i}$
  - ELSE
    - $\mathbf{x}_k^i = -\sqrt{\mathbf{s}_k^i}$
  - END IF
  - $w_k^i = w_{k-1}^i p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)\mathbf{x}_k^i$
- 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
- Calculate $\widehat{N_{eff}}$ using (51)
- 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

References

[1] 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.

[2] Y Bar-Shalom & X Rong Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, 1995.

[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 non-linear 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 Non-linear 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 Non-linear Filtering Problems. *Annals of Applied Probability*, 1998, Volume 8, No2, pp 438-495.

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

[11] P Del Moral, Non-linear 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 Non-linear and Non-Gaussian Bayesian State Estimation. *IEE Proceedings-F*, 1993, Volume 140, pp.107-113.

[18] G D Forney, The Viterbi Algorithm, *Proc. of the IEEE*, March 1973, Volume 61(3), pp.268-278.

[19] 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.

[20] R E Helmick, D Blair & S A Hoffman, Fixed-Interval Smoothing for Markovian Switching Systems. *IEEE Transactions on Information Theory*, 1995, Volume 41(6), pp.1845-1855.

[21] Y C Ho & R C K Lee, A Bayesian Approach to Problems in Stochastic Estimation and Control. *IEEE Trans. Auto. Control*, 1964, Volume AC(9), pp.333-339.

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

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

[24] 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.

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

[26] G Kitagawa & W Gersch, *Smoothness priors analysis of time series*, Springer-Verlag, New York, 1996.

[27] 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.

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

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

[30] F Martinerie & P Forster, Data Association and Tracking Using Hidden Markov Models and Dynamic Programming. *Proc. Conf. ICASSP 92*, 1992.

[31] 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.

[32] M Orton & A Marrs, Particle Filters for Tracking with Out-Of-Sequence Measurements. To be submitted to IEEE Transactions.

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

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

[35] L R Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc of the IEEE*, February 1989, Volume 77(2), pp.257-285.

[36] L R Rabiner & B H Juang, An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, January 1986, pp.4-16.

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

[38] R H Shumway & D S Stoffer, An Approach to Time Series Smoothing and Forecasting Using the EM Algorithm. *Journal of Time Series Analysis*, 1982, Volume 3(4), pp.253-264.

[39] R L Streit & R F Barrett, Frequency Line Tracking Using Hidden Markov Models. *IEEE Trans. Accoust. Speech & Sig. Proc.*, April 1990, Volume 38(4), pp.586-598.

[40] 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.

[41] M West & J Harrison, Bayesian Forecasting and Dynamic Models. *Springer Series in Statistics*, 2nd Edition, 1997, Springer-Verlag, New York.

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

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

Sanjeev Arulampalam received the B.Sc degree in Mathematical Sciences and the B.E degree with first class honours in Electrical and Electronic Engineering from the University of Adelaide in 1991 and 1992, respectively. In 1992 he joined the staff of Computer Sciences of Australia (CSA) where he worked as a Software Engineer in the Safety Critical Software Systems group. In 1993, he won a Telstra Postgraduate Fellowship award to work toward a Ph.D degree in Electrical and Electronic Engineering at the University of Melbourne, which he completed in 1997. His doctoral dissertation was "Performance Analysis of Hidden Markov Model based Tracking Algorithms". Upon completion of his postgraduate studies, in 1998 Dr Arulampalam joined the Defence Science and Technology Organisation (DSTO), Australia, as a Research Scientist in the Surveillance Systems Division. Here he carried out research in many aspects of airborne target tracking with a particular emphasis on tracking in the presence of deception jamming. In 2000, he won the Anglo-Australian postdoctoral fellowship, awarded by the Royal Academy of Engineering, London. This postdoctoral research was carried out in the U.K, both at the Defence Evaluation and Research Agency (DERA) and at Cambridge University. His research interests include estimation theory, target tracking and sequential Monte-Carlo methods.

Simon Maskell received his BA degree in Engineering and an MEng in Electronic and Information Sciences from Cambridge University Engineering Department, CUED, both in 1999. He works in the Pattern and Information Processing group at QinetiQ Ltd. In 2000, he was awarded a Royal Commission for the Exhibition of 1851 Industrial Fellowship and as a result is currently a PhD student at CUED. His research interested include Bayesian inference, signal processing, tracking and data fusion with particular emphasis on the application of particle filters.

Neil Gordon obtained a BSc in Mathematics and Physics from Nottingham University in 1988 and a PhD degree in statistics from Imperial College, University of London in 1993. He is currently with the Pattern and Information Processing group at QinetiQ Ltd. His research interests include Bayesian estimation and sequential Monte Carlo methods (aka particle filters) with a particular emphasis on target tracking and missile guidance. He has co-edited, with A Doucet and JFG de Freitas, Sequential Monte Carlo methods in practice (New York: Springer-Verlag).

Tim Clapp received his B.A. and M.Eng. degrees from Cambridge University in 1996 before gaining his Ph.D. in the Signal Processing and Communications Group in Cambridge University Engineering Department. His research interests include blind equalisation, Markov chain Monte Carlo techniques and particle filters. He currently works on Telecommunications satellite system design for the Payload Processor Group at Astrium Ltd.