

A Survey of Numerical Methods for Nonlinear Filtering Problems.*

Amarjit Budhiraja[†]
Department of Statistics and Operations Research
University of North Carolina
Chapel Hill, NC 27599

Lingji Chen
Scientific Systems Company Inc.
Woburn, MA 01801

Chihoon Lee
Department of Statistics and Operations Research
University of North Carolina
Chapel Hill, NC 27599

September 4, 2005

Abstract

The main goal of filtering is to obtain, recursively in time, good estimates of the state of a stochastic dynamical system based on noisy partial observations of the same. In settings where the signal/observation dynamics are significantly nonlinear or the noise intensities are high, extended Kalman filter(EKF), which is essentially a first order approximation to an infinite dimensional problem, can perform quite poorly: it may require very frequent re-initializations and in some situations may even “blow up”. The theory of nonlinear filtering addresses these difficulties by considering the evolution of the conditional distribution of the state of the system given all the available observations, in the space of probability measures. We survey a variety of numerical schemes that have been developed in the literature for approximating the conditional distribution described by such stochastic evolution equations; with a special emphasis on an important family of schemes known as the particle filters. A numerical study is presented to illustrate that in settings where the signal/observation dynamics are non linear a suitably chosen nonlinear scheme can drastically out perform the extended Kalman filter.

*Research of all three authors was supported in part by Army STTR Phase II contract W911NF-04-C-0108

[†]Research supported in part by ARO (Grant W911NF-04-1-0230)

1 Background.

One of the key goals of data assimilation is to obtain, recursively in time, good estimates of the state of a stochastic dynamical system based on noisy partial observations of the same. The major breakthrough in this classical problem of signal analysis was the landmark work of Kalman and Bucy[14]. The power of Kalman filter, which can be gauged from the broad spectrum of disciplines where it is used, lies in its elegant simplicity. Under the standard assumptions of linearity and Gaussianity one has that the conditional distribution of the state of the system at time t , given all the observations up until time t is Gaussian. By standard assumptions we mean that the state process (also referred to as the signal), denoted by $\{X_t; t \geq 0\}$ is a linear stochastic dynamical system driven by a Gaussian white noise (\dot{W}_t); (X_0, W) is jointly Gaussian and the observations, $\{Y_t; t \geq 0\}$, have a linear relationship with the state variable, up to a measurement error term which is given by a Gaussian white noise (\dot{B}_t) independent of (X_0, W) . The conditional distribution gives the best estimate of the statistical description of the state of the system (in the least square sense) based on all measurements available up to the current time. Since the conditional distribution is Gaussian, it is parametrized by only two sets of parameters: mean vector \hat{X}_t and the covariance matrix P_t . Kalman filter gives a linear SDE (stochastic differential equation) for the mean and a Ricatti type ODE (ordinary differential equation) for the variance. The ODE can be solved off-line and the SDE being a linear one, has several efficient numerical algorithms for its approximation.

The first problem one encounters when one goes from theory to real situation applications of filtering ideas is that very often the assumption of linearity is violated. For example, in typical atmospheric data assimilation problems, the signal dynamics, which is given via a rather complex system of equations from fluid mechanics, is very nonlinear. Furthermore, depending on how the measurements are obtained, the observation process can also have a nonlinear relation with the state process. Although the conditional distribution still captures the best statistical description of the state given the observations, in a general nonlinear setting, it typically fails to be Gaussian. Furthermore, excepting some special cases (cf. Beneš[1]), the evolution of the conditional distribution cannot be captured by that of a finite set of parameters. The theory of nonlinear filtering (cf. Kallianpur[13], Kushner[17], Liptser-Shiryaev[20]), addresses these difficulties by considering the evolution of the conditional distribution in the space of probability measures. These stochastic evolutions descriptions for the nonlinear filter Π_t (i.e. the conditional distribution), the so called Zakai's equation and Kushner-Stratonovich equation, are some of the key results in this area. Before the advent of modern computing technology the implementation of the solution to the general nonlinear filtering problem by numerical approximations of such infinite dimensional stochastic evolution equations seemed impractical. Therefore, algorithms which used linear approximations to the signal-observation dynamics were studied, leading to the development of the extended Kalman filter(EKF) (cf. [12]). In settings where the dynamics are significantly nonlinear or the noise intensities are high, EKF, which is essentially a first order approximation to an infinite dimensional problem, can perform quite poorly: it may require very frequent re-initializations and in some situations may even “blow up” (see Section 5 for an example).

In view of these difficulties and tremendous computational resources available currently it becomes of great interest to revisit the general solution of the nonlinear filtering problem and examine the feasibility and complexity/accuracy tradeoff for various numerical schemes for approximating

the nonlinear filter. The goal of this survey is to present a variety of numerical schemes that have been studied in the literature with a special emphasis on a very attractive class of approximation methods, the so called particle filters.

2 Problem Formulation.

We will denote the state of the unobserved stochastic dynamical system at a time instant t by $X(t)$ and the (integrated) observations at time t by $Y(t)$. Although the general formulation of the nonlinear filtering problem requires very little structure, other than the Markov property, on the stochastic processes (X, Y) ; for the sake of specificity we will assume that X is a k dimensional diffusion satisfying

$$dX(t) = f(X(t))dt + \sigma(X(t))dW(t), \quad (2.1)$$

where W is a p dimensional standard Wiener process and $f : \mathbb{R}^k \rightarrow \mathbb{R}^k$, $\sigma : \mathbb{R}^k \rightarrow \mathcal{M}(k \times p)$ satisfy appropriate regularity conditions. Here $\mathcal{M}(k \times p)$ is the space of $k \times p$ matrices. The observation process is a \mathbb{R}^m valued continuous stochastic process given via the equation

$$Y(t) = \int_0^t g(X(s))ds + B(t), \quad (2.2)$$

where $g : \mathbb{R}^k \rightarrow \mathbb{R}^m$ is a suitable function and B is a standard m dimensional Wiener process that is independent of (X_0, B) . The goal of nonlinear filtering is the study of the conditional distribution of $X(t)$ given all the observations up until time instant t , i.e. $\{Y(s) : 0 \leq s \leq t\}$. This conditional distribution, which is a (random) probability measure, is referred to as the nonlinear filter and will be denoted by $\Pi(t)$. The two main challenges in the computation of the nonlinear filter are:

(1) **Online implementation and explosion of data.** The main application of filtering is real time tracking which means that filtering estimates need to be produced within a very short time span. Furthermore, as time evolves more and more observations become available leading to serious data storage issues. In view of these features it becomes critical that the filter be computed recursively in time. Namely, the computation of the filter at time $t + h$ should only require the filter at time t and observations collected over $[t, t + h)$ and not the whole observation history.

(2) **Infinite dimensionality.** The filtering problem is inherently infinite dimensional. Namely, even if one is interested in just computing the posterior means and variances, one needs to work with recursive equations for the whole conditional probability distribution.

Below we provide two infinite dimensional evolution equations for the nonlinear filter which suggest a recursive way for computing the nonlinear filter. In practice, of course, these evolution equations need to be discretized using suitable numerical approximation schemes.

2.1 Kallianpur-Striebel Formula.

In order to motivate this formula we consider a simple discrete time filtering problem where the signal $\{X_n\}$ is a \mathbb{R} valued Markov chain with a transition density $p(x, y)$. Also, let the observations

be given by the equation: $Y_n = g^*(X_n) + \xi_n$, where ξ_n is i.i.d. $N(0,1)$ and g^* is a continuous function. In this problem, the filter, Π_n (i.e. the conditional distribution of the signal at instant n given the observations up to time n) admits a density denoted as $f_{X_n|Y_1, \dots, Y_n}$. Using the Bayes rule one has that

$$f_{X_n|Y_1, \dots, Y_n}(x) = \frac{f_{X_n, Y_1, \dots, Y_n}(x, Y_1, \dots, Y_n)}{f_{Y_1, \dots, Y_n}(Y_1, \dots, Y_n)},$$

where we have used the conventional notation for joint and conditional densities. The numerator above can be rewritten as $\int f_{X_{n-1}, X_n, Y_1, \dots, Y_n}(\tilde{x}, x, Y_1, \dots, Y_n) d\tilde{x}$. One can rewrite the denominator similarly. Next, using the Markov property it is easily seen that

$$f_{X_{n-1}, X_n, Y_1, \dots, Y_n}(\tilde{x}, x, Y_1, \dots, Y_n) = c f_{X_{n-1}|Y_1, \dots, Y_{n-1}}(\tilde{x}) f_{X_n|X_{n-1}=\tilde{x}}(x) f_{Y_n|X_n=x}(Y_n),$$

where c is a normalization factor depending only on Y_1, \dots, Y_n . After some simplifications, in particular noting that $f_{Y_n|X_n=x}(Y_n) = c e^{-(g^*(x)-Y_n)^2/2}$, one sees that for a suitable test function ϕ

$$\int \phi(x) \Pi_n(dx) = \frac{\mathbb{E}_{\{\Pi_{(n-1)}, Y_n\}} \left(\phi(\tilde{X}_1) e^{-\frac{1}{2}(g^*(\tilde{X}_1)-Y_n)^2} \right)}{\mathbb{E}_{\{\Pi_{(n-1)}, Y_n\}} \left(e^{-\frac{1}{2}(g^*(\tilde{X}_1)-Y_n)^2} \right)}.$$

Here $\{\tilde{X}_n\}$ is a Markov chain, independent of $\{X_n, \xi_n\}$ with transition density p and initial condition Π_{n-1} . The above expectations are taken conditional on the observation Y_n and the initial data Π_{n-1} .

Kallianpur-Streibler formula provides an extension of the above representation formula (which is essentially obtained via Bayes rule and application of the Markov property) to the continuous time framework. Consider now the filtering problem for the continuous time stochastic processes $(X(t), Y(t))$ given by equations (2.1), (2.2). Then for suitable test functions $\phi : \mathbb{R}^k \rightarrow \mathbb{R}$, this representation formula is as follows.

$$\int \phi(x) \Pi(t)(dx) = \frac{\mathbb{E}_{\{\Pi(t-s), Y_{t-s,t}\}} \left[\phi(\tilde{X}(s)) R(\tilde{X}_{0,s}, Y_{t-s,t}) \right]}{\mathbb{E}_{\{\Pi(t-s), Y_{t-s,t}\}} R(\tilde{X}_{0,s}, Y_{t-s,t})}, \quad 0 \leq s \leq t \quad (2.3)$$

where

$$R(\tilde{X}_{0,s}, Y_{t-s,t}) = \exp \left[\int_0^s g'(\tilde{X}(u)) d_u Y(t-s+u) - \frac{1}{2} \int_0^s |g(\tilde{X}(u))|^2 du \right],$$

\tilde{X} is independent of X with the same transition law as X and initial distribution $\Pi(t-s)$. The notation $\mathbb{E}_{\pi, Y_{t-s,t}}$ denotes the expectation conditioned on the data $Y_{t-s,t} \doteq \{Y(t-s+u) - Y(t-s) : 0 \leq u \leq s\}$ and π being the initial distribution of \tilde{X} . The above formula says that the optimal filter $\Pi(t)$ can be obtained recursively as follows. Having obtained the optimal filter at time instant $t-s$, i.e. $\Pi(t-s)$, the filter at time t can be obtained by evaluating the expectations in the numerator and denominator in (2.3), in which the observation path $Y_{t-s,t}$ is treated as a fixed given trajectory.

2.2 Zakai's Equation.

An alternative approach to the computation of the nonlinear filter is by developing an evolution equation for the numerator of the term on the right side of (2.3):

$$\rho_t(\phi) \doteq \mathbb{E}_{\{\Pi(0), Y_{0,t}\}} \left(\phi(\tilde{X}(t)) R(\tilde{X}_{0,t}, Y_{0,t}) \right).$$

This leads to the well known stochastic integro-differential equation, called the Zakai's equation.

$$d\rho_t(\phi) = \rho_t(L\phi)dt + \rho_t(gf)dY_t, \quad (2.4)$$

where L is the infinitesimal generator of the diffusion X . Under suitable regularity conditions one can show that the random measure ρ_t admits a density $u(t, x)$ (for Kalman filter this is a Gaussian density, up to normalization), i.e. $\rho_t(A) = \int_A u(t, x)dx$, $A \subset \mathbb{R}^k$, which is the unique solution of the following stochastic partial differential equation (SPDE):

$$du(t, x) = L^*u(t, x)dt + h(x)u(t, x)dY(t), \quad (2.5)$$

where L^* is the (formal) adjoint of L . The above SPDE gives a recursive way for evaluating $u(t, x)$ and there have been several works [3, 4, 21, 2] which have developed suitable time and space discretization schemes for this SPDE in order to obtain numerical approximations for the nonlinear filter. Note that all the needed information about the state at time t can be obtained from $u(t, x)$ via suitable integrations. For example the conditional expectation of $\phi(X(t))$ given all the observations up until t is given as $\int \phi(x)u(t, x)dx / \int u(t, x)dx$.

2.3 Discrete Observations.

In typical applications, observations are available only at a discrete sequence of time instants. We will assume here that the observations are collected at time instants $h, 2h, \dots$. Denoting $Y_n \doteq Y(nh) - Y((n-1)h)$ we have

$$Y_n = \int_{(n-1)h}^{nh} g(X(s))ds + W(nh) - W((n-1)h) \approx g(X(nh))h + \theta\xi_n,$$

where $\xi_n \doteq h^{-1/2}(W(nh) - W((n-1)h))$ is an i.i.d. sequence of standard Gaussian variables and $\theta = \sqrt{h}$. Denoting $hg(x)$ by $g^*(x)$ we can write a discrete analog of the observation equation (2.2):

$$Y_n = g^*(X(nh)) + \theta\xi_n, \quad n \geq 1. \quad (2.6)$$

Alternatively, one can view (2.6) as the original underlying observation model and not a discrete approximation to (2.2).

The nonlinear filter in this discrete observation setting is a sequence $\{\Pi_n\}$ of probability measure valued random variables, where Π_n is the conditional distribution of $X(nh)$ given the observations Y_1, \dots, Y_n . As noted earlier, the Kallianpur-Striebel formula in this setting simply reduces to a Bayes formula of the form given as follows: For suitable test functions ϕ

$$\int \phi(x)\Pi_n(dx) = \frac{\mathbb{E}_{\{\Pi_{(n-1)}, Y_n\}} \left[\phi(\tilde{X}(h)) \tilde{R}(\tilde{X}(h), Y_n) \right]}{\mathbb{E}_{\{\Pi_{(n-1)}, Y_n\}} \tilde{R}(\tilde{X}(h), Y_n)}. \quad (2.7)$$

where

$$\tilde{R}(x, y) = \exp \left[\frac{g^*(x) \cdot y}{\theta^2} - \frac{|g^*(x)|^2}{2\theta^2} \right]$$

and the process \tilde{X} is a diffusion solving (2.1) with W replaced by a standard Brownian motion \tilde{W} that is independent of $(W, B, X(0))$ and $\tilde{X}(0)$ is independent of all other processes and has the distribution Π_{n-1} . Once again the formula can be interpreted as a recursive algorithm for computing the nonlinear filter, as follows. Having obtained the nonlinear filter at time instant $n-1$, the filter at instant n can be obtained by evaluating the above expectations where Y_n is held fixed and \tilde{X} is independent of X with same transition law as X and initial distribution Π_{n-1} . The Zakai's equation in the discrete observation set up reduces to a recursive relation of the following form.

$$\Pi_n(dx) = c\tilde{R}(x, Y_n) \int G(\tilde{x}, dx)\Pi_{n-1}(d\tilde{x}), \quad (2.8)$$

where $G(\tilde{x}, dx) = \mathbb{P}(X(h) \in dx \mid X(0) = \tilde{x})$ is the h -step transition kernel of the Markov diffusion process X .

2.4 Two key steps of the recursion.

Equation (2.8) suggests a natural way to divide the computation of Π_n from Π_{n-1} into two main steps, given as follows.

Prediction step. This corresponds to computing the best estimate of the state at time nh given all the observations Y_1, \dots, Y_{n-1} , but not Y_n . Schematically this step can be represented as

$$\Pi_{n-1}(dx) \mapsto \hat{\Pi}_n(dx) = \int G(\tilde{x}, dx)\Pi_{n-1}(d\tilde{x}). \quad (2.9)$$

Here $\hat{\Pi}_n$ represents the conditional distribution of $X(nh)$ given Y_1, \dots, Y_{n-1} . The key ingredients in computing $\hat{\Pi}_n$ from Π_{n-1} are the computation of the kernel $G(\tilde{x}, dx)$ and that of the (multi-dimensional) integral on the right side of (2.9). Note that even in settings where the transition kernel is explicitly known the computation of the integral when the state dimension is high is quite nontrivial. Furthermore, for diffusion type models as considered in the current work, $G(\tilde{x}, dx)$ is given as the fundamental solution of the Fokker-Planck equation associated with the elliptic operator L^* . Numerical solutions of such PDEs get prohibitively expensive as the state dimension increases.

Filtering step. The second main step in the recursion (2.8) is the assimilation of the observation at instant n , i.e. Y_n , into the predictor $\hat{\Pi}_n$ to obtain the best estimate of the distribution of $X(nh)$ given Y_1, \dots, Y_n , i.e. Π_n . This step can be schematically expressed as

$$\hat{\Pi}_n(dx) \mapsto \Pi_n(dx) = c\tilde{R}(x, Y_n)\hat{\Pi}_n(dx), \quad (2.10)$$

where c is the normalization constant. The most expensive part in the filtering step is the computation of the normalization factor $c = \int \tilde{R}(x, Y_n)\hat{\Pi}_n(dx)$. Once again, as the state dimension increases, the computation of this multidimensional integral becomes significantly harder.

3 Approximations to the Nonlinear Filter.

We will consider in this section a variety of approximation schemes for the optimal nonlinear filter. A common feature of all these schemes is that each of them can be broadly divided into two steps: (1) Approximation of the prediction step; (2) Approximation of the filtering step. One reason for distinguishing between these two steps is that typically in a given application one would like to “mix and match” to obtain an algorithm which uses one scheme for the prediction step and perhaps some other scheme for the filtering step. For example, in typical data assimilation problems, where observations are linear, one may wish to use a Kalman filter for the filtering step while a suitable nonlinear scheme for the prediction step. We begin with the most commonly followed approach for numerics of a nonlinear filtering problem namely the Extended Kalman Filter.

3.1 Extended Kalman Filter (cf. [12]).

The basic idea here is to linearize the dynamics and as a result approximate the conditional distribution Π_n by a normal distribution with parameters (m_n, P_n) . Given the approximation Π_{n-1}^* to Π_{n-1} as a normal distribution with mean m_{n-1} and covariance P_{n-1} , the approximation Π_n^* to Π_n is obtained as follows.

Approximating the prediction step. Let $\{x^*(t), t \geq 0\}$ be the trajectory about which linearization is performed. Although there is a lot of flexibility in the choice of x^* , typically it is chosen as the solution of the ODE $\dot{x}(t) = f(x(t))$ with initial condition m_{n-1} . Letting $\Delta X(t) = X(t) - x^*(t)$, the linearized signal dynamics is given as

$$\Delta \dot{X}(t) = \left[\frac{\partial f}{\partial x} \right]_{x=x^*(t)} \Delta X(t) + \sigma(x^*(t)) dB(t).$$

The prediction step of the usual (continuous time) Kalman filter is then used with this modified linearized dynamics to give the approximation $\hat{\Pi}_n^*$ to $\hat{\Pi}_n$ via a normal distribution with parameters (\hat{m}_n, \hat{P}_n) .

Approximating the filtering step. The linearized observation model is given as follows.

$$Y_n - g^*(\hat{m}_n) = \left[\frac{\partial g^*}{\partial x} \right]_{x=\hat{m}_n} (X(nh) - \hat{m}_n) + \theta \xi_n.$$

The filtering step of the usual (discrete time) Kalman filter is used with this linearized observation model to obtain Π_n^* as a normal distribution with parameters (m_n, P_n) . The cycle can now be repeated with n replaced by $n + 1$.

3.2 Moment Closing Algorithm (cf. [16, 5]).

The approach here is once again to approximate the nonlinear filter by a Gaussian distribution; however instead of directly linearizing the dynamics one tries to match the moments of the of the optimal filter with that of a Gaussian random variable. Given the approximation Π_{n-1}^* to Π_{n-1}

as a normal distribution with mean m_{n-1} and covariance P_{n-1} , the approximation Π_n^* to Π_n is obtained as follows.

Approximation to the prediction step. Denote a Gaussian density with parameters (m, P) by $\mathcal{N}(m, P)$. Choose $\delta \ll h$ such that $h/\delta \doteq \kappa$ is an integer. Define for $i = 1, \dots, \kappa$

$$\eta_i \doteq \eta_{i-1} + \delta f(\eta_{i-1}) + \sqrt{\delta} \sigma(\eta_{i-1}) \gamma_i$$

where $\{\gamma_i\}$ is an i.i.d. sequence of standard normal vectors and η_0 is normally distributed with parameters (m_{n-1}, P_{n-1}) and is independent of $\{\gamma_i\}$. We approximate the law of each η_i by a Gaussian distribution with a density denoted by p_i . The approximation Π_n^* to $\hat{\Pi}_n$ is then given by p_κ . The parameters of this density are determined as follows. Note that $p_0 = \mathcal{N}(m_{n-1}, P_{n-1})$. Suppose now that $p_i = \mathcal{N}(\tilde{m}_i, \tilde{P}_i)$. Then

$$\tilde{m}_{i+1} \approx \mathbb{E}(\eta_{i+1}) \approx \tilde{m}_i + \delta \int_{\mathbb{R}^k} f(x) \mathcal{N}(\tilde{m}_i, \tilde{P}_i)(x) dx$$

and denoting $\sigma\sigma^*$ by a ,

$$\begin{aligned} \tilde{P}_{i+1} &\approx \text{Var}(\eta_i + \delta f(\eta_i)) + \delta \mathbb{E}(a(\eta_i)) \\ &\approx \int_{\mathbb{R}^k} \left((x + f(x)\delta - \tilde{m}_{i+1})(x + f(x)\delta - \tilde{m}_{i+1})^* + \delta a(x) \right) \mathcal{N}(\tilde{m}_i, \tilde{P}_i)(x) dx. \end{aligned}$$

The above integrals are approximated using a suitable numerical integration scheme such as Gaussian Quadrature. Often a pre-computation can be done to simplify the work and reduce the dimension of the space over which numerical integration is performed. In particular if σ is deterministic and f has a linear relationship with a certain subset of variables then these variables can be eliminated from the state space over which numerical integration is done.

Approximation to the filtering step. Given the approximation $\hat{\Pi}_n^*$ as a Normal distribution with parameters (\hat{m}_n, \hat{P}_n) , the Gaussian approximation $\Pi_n^* \sim \mathcal{N}(m_n, P_n)$ to Π_n is obtained as follows.

$$\begin{aligned} m_n &= c_n^{-1} \int_{\mathbb{R}^k} x I_n(x) \mathcal{N}(\hat{m}_n, \hat{P}_n)(x) dx, \\ P_n &= c_n^{-1} \int_{\mathbb{R}^k} (x - m_n)(x - m_n)^* I_n(x) \mathcal{N}(\hat{m}_n, \hat{P}_n)(x) dx, \end{aligned}$$

where c_n is the normalization constant $\int_{\mathbb{R}^k} I_n(x) \mathcal{N}(\hat{m}_n, \hat{P}_n)(x) dx$ and $I_n(x) \doteq e^{-\frac{1}{2\theta^2}(Y_n - g^*(x))'(Y_n - g^*(x))}$. The above integrations, once more, are approximated by a suitable numerical integration scheme such as Gaussian Quadrature.

3.3 Markov Chain Approximation (cf. [17]).

It is well known that the Kalman filter suffers from serious divergence and instability problems when the deviations from linearity in the signal/observation dynamics are significant or signal to noise ratio is low. Furthermore, a Gaussian approximation to the nonlinear filter fails to capture the multi-modal features of the true conditional distribution. An alternative approach to the

approximation problem is to use the representation for the nonlinear filter given by the Kallianpur-Striebel formula (2.7) and approximate the expectations in the numerator and denominator of its right hand side in a suitable manner. As noted earlier, computing these expectations, which are given in terms of the diffusion \tilde{X} , involves solving certain Fokker-Planck equations. So a natural approach is to replace the process \tilde{X} by a simpler process X^* which well approximates \tilde{X} and is such that the corresponding expectations are easier to compute. This is the basic motivation for the Markov chain approximation method which uses a finite state discrete time Markov Chain X^* as an approximation for \tilde{X} . Roughly speaking, this corresponds to taking a finite difference approximation for the associated Fokker-Planck equation. The basic scheme is as follows. Recall the h step transition kernel G introduced below (2.8). Suppose for this discussion that the state space is a compact subset \mathbb{S} of \mathbb{R}^k . Let $\delta \ll h$ and define $\mathbb{S}_\delta \doteq \mathbb{S} \cap \delta\mathbb{Z}^k$. Let G_δ be a transition kernel on \mathbb{S}_δ such that G_δ “approaches” G as $\delta \rightarrow 0$. We refer the reader to [18] for details on construction of G_δ for jump-diffusion type models. Given an approximation Π_{n-1}^* of Π_{n-1} as a discrete measure supported on \mathbb{S}_δ , the two key steps of filtering are carried out as follows.

- Prediction step reduces to a matrix multiplication:

$$\hat{\Pi}_n^*(x) = \sum_{\tilde{x} \in \mathbb{S}_\delta} G_\delta(\tilde{x}, x) \Pi_{n-1}^*(\tilde{x}), x \in \mathbb{S}_\delta.$$

- Filtering step is given as

$$\Pi_n^*(x) = c\tilde{R}(x, Y_n)\hat{\Pi}_n^*(x), x \in \mathbb{S}_\delta,$$

where c is a normalization constant. Compactness of the state space essentially amounts to using a fixed grid for the numerical approximation to the associated Fokker-Planck equation. In practice, however, one uses the information from the filtering results to dynamically move the grid in a suitable manner. The method works very well in 2-3 dimensional signal models, however it becomes computationally very expensive in higher dimensions.

4 Particle Filters.

In high dimensional problems particle filters (cf. [15, 7, 8, 11, 6]) are an attractive alternative to numerical approximation of the Fokker-Planck equations by finite difference or finite element methods. These methods have a long history; we refer the reader to [10] for an extensive treatment of this area. The basic idea is to replace the expected values in (2.7) by suitable Monte-Carlo sample averages. An appealing aspect of these algorithms is that unlike the Markov chain approximation method or any other standard discretization scheme for Fokker-Planck equations, one does not need to fix a grid to approximate the dynamics. Particle methods are very flexible and easy to implement; also they are ideally suited for a parallel computing architecture. We will now give a survey of a variety of particle filters that have been studied in the literature.

4.1 A basic form of particle filter.

The approximation Π_n^* to Π_n is given by a discrete probability measure supported on points $x_1^{(n)}, \dots, x_L^{(n)}$ and corresponding weights $p_1^{(n)}, \dots, p_L^{(n)}$. Here L represents the number of particles

that are used to approximate the distribution Π_n . Given the approximation Π_{n-1}^* to Π_{n-1} by the discrete probability measure $\{(x_1^{(n-1)}, p_1^{(n-1)}), \dots, (x_L^{(n-1)}, p_L^{(n-1)})\}$ the two key steps of the algorithm are as follows.

Prediction step. Propagate each of the particles $x_j^{(n-1)} \mapsto \hat{x}_j^{(n)}$ using the signal dynamics. This requires simulating an approximate solution of the SDE (2.1) with initial condition $x_j^{(n-1)}$ by some discretization scheme. For example, in an Euler scheme one fixes a $\delta \ll h$ such that $h/\delta \doteq \kappa$ is an integer and obtains, iteratively,

$$\eta_i \doteq \eta_{i-1} + f(\eta_{i-1})\delta + \sqrt{\delta}\sigma(\eta_{i-1})\gamma_i$$

where $\{\gamma_i\}$ is an i.i.d. sequence of standard normal vectors and $\eta_0 = x_j^{(n-1)}$. The propagated point $\hat{x}_j^{(n)}$ is then set equal to η_κ . This gives an approximation $\hat{\Pi}_n^*$ to $\hat{\Pi}_n$ as the discrete probability distribution $\{(\hat{x}_1^{(n)}, \hat{p}_1^{(n)}), \dots, (\hat{x}_L^{(n)}, \hat{p}_L^{(n)})\}$ where we set $\hat{p}_j^{(n)} \doteq p_j^{(n-1)}$.

Filtering step. Update the weights $\hat{p}_j^{(n)} \mapsto p_j^{(n)}$ using the observation Y_n by setting $p_j^{(n)} \doteq cp_j^{(n-1)} \tilde{R}(\hat{x}_j^{(n)}, Y_n)$, where c is the normalization constant. The approximation Π_n^* is now given as $\{(x_1^{(n)}, p_1^{(n)}), \dots, (x_L^{(n)}, p_L^{(n)})\}$ where we set $x_j^{(n)} \doteq \hat{x}_j^{(n)}$. Although the scheme is very easy to implement, it suffers from severe degeneracy problems, especially in high dimensions. The main difficulty is that after a few time steps all the weights tend to concentrate on a very few particles which drastically reduces the effective sample size. A common remedy for this paucity of significant particles is to occasionally re-sample in order to rejuvenate the particle cloud.

4.2 Particle filter with occasional resampling.

The main idea here is to periodically resample with replacement from the discrete distribution approximating the filter to obtain a uniform distribution of weights. Of course, resampling adds extra noise to the approximating scheme so it is important not to resample too frequently. The precise algorithm is as follows. Let δ, κ, L be as before. Fix a resampling lag parameter $\alpha \in \mathbb{N}$. Suppose that an approximation Π_{n-1}^* as a discrete distribution $\{(x_1^{(n-1)}, p_1^{(n-1)}), \dots, (x_L^{(n-1)}, p_L^{(n-1)})\}$ is given. Then the recursion is given as follows. Suppose first that n/α is not an integer. In this case the approximation $\Pi_n^* = \{(x_1^{(n)}, p_1^{(n)}), \dots, (x_L^{(n)}, p_L^{(n)})\}$ is given exactly as before in Section 4.1. If n/α is an integer we further modify the above discrete probability measure Π_n^* as follows. Take a random sample of size L , with replacement, from the discrete distribution $\{(x_1^{(n)}, p_1^{(n)}), \dots, (x_L^{(n)}, p_L^{(n)})\}$. Relabel the new points as $x_1^{(n)}, \dots, x_L^{(n)}$. The approximation Π_n^* is then given as the discrete distribution: $\Pi_n^* = \{(x_1^{(n)}, p_1^{(n)}), \dots, (x_L^{(n)}, p_L^{(n)})\}$ where all the $p_i^{(n)}$ are set equal to $1/L$.

4.3 Variance reduction: Deterministic allocation and residual resampling.

As noted earlier, one drawback of the above algorithm is that it needlessly introduces extra variability in the algorithm due to random sampling. To deal with this one usually conducts the resampling step using some variance reduction scheme. We describe below one commonly used

scheme. Another variance reduction scheme is given in the next subsection. Let δ, κ, L and α be as in Section 4.2. The key difference here is that at a resampling step, i.e. when n/α is an integer, instead of random sampling with replacement from $\{(x_1^{(n)}, p_1^{(n)}), \dots, (x_L^{(n)}, p_L^{(n)})\}$ we do a (partial) deterministic allocation as follows. Let $n_i \doteq \lfloor L p_i^{(n)} \rfloor$ and branch $x_i^{(n)}$ into n_i particles. This yields $\sum_{i=1}^L n_i$ particles which is less than or equal to L . Set $L_{\text{res}} \doteq L - \sum_{i=1}^L n_i$ and set $w_n^{(i)} \doteq L p_i^{(n)} - n_i$. Next, resample L_{res} particles using random sampling with replacement from the distribution $\{(x_1^{(n)}, c w_1^{(n)}), \dots, (x_L^{(n)}, c w_L^{(n)})\}$, where c is a normalizing constant. This gives a total of L particles which are relabeled as $x_1^{(n)}, \dots, x_L^{(n)}$. Finally, as before, define the approximation Π_n^* to Π_n as $\{(x_1^{(n)}, p_1^{(n)}), \dots, (x_L^{(n)}, p_L^{(n)})\}$ where all the $p_i^{(n)}$ are set equal to $1/L$.

4.4 Variance reduction scheme: A branching particle filter (cf. [9, 8]).

In this variation of a variance reduction method the number of particles is allowed to change at each time step. Let $L, \delta, \kappa, \alpha$ be as before. The main steps are as follows. Suppose that an approximation Π_{n-1}^* is given by $L^{(n-1)}$ points $x_j^{(n-1)}, j = 1, \dots, L^{(n-1)}$ with equal weights. Propagate these points forward for time length h using numerical SDE solvers for equation (2.1) as in Section 4.1. Call the new points $\hat{x}_j^{(n)}, j = 1, \dots, L^{(n-1)}$. The main difference from Sections 4.2, 4.3 is that instead of reweighing the particles as in the above algorithm we “branch and kill particles”. More precisely, if n/α is not an integer, a new cloud of particles (starting from the particles $\hat{x}_j^{(n)}$) is obtained as follows. The j -th particle branches (independently of other particles) into a random number (denoted by ζ_j^n) of particles, for $j = 1, \dots, L^{(n-1)}$. The distributions of ζ_j^n are given as follows.

$$\begin{aligned} P(\zeta_j^n = [\gamma_j^n] + 1) &= \gamma_j^n - [\gamma_j^n], \\ P(\zeta_j^n = [\gamma_j^n]) &= 1 - \gamma_j^n + [\gamma_j^n], \end{aligned}$$

where $\gamma_j^n \doteq \frac{L^n \beta_j^n}{\sum_{k=1}^{L^n} \beta_k^n}$. and $\beta_j^n \doteq \tilde{R}(x_j^n, Y_n)$. This results in say a total of L^n particles which are denoted once more by $\{x_j^n; 1 \leq j \leq L^n\}$. The approximation Π_n^* is then given as $\{(x_1^{(n)}, p_1^{(n)}) \cdots (x_{L^{(n)}}^{(n)}, p_{L^{(n)}}^{(n)})\}$ where each $p_i^{(n)}$ is set equal to $1/L^{(n)}$. In order to manage the explosion or decay of the number of particles $L^{(n)}$ we add a resampling step at every α time steps. More precisely, if n/α is an integer, we sample (with replacement) L points from the distribution $\{(x_1^{(n)}, p_1^{(n)}) \cdots (x_{L^{(n)}}^{(n)}, p_{L^{(n)}}^{(n)})\}$. Relabeling the new points once more as $\{x_j^{(n)}\}$, the approximation Π_n^* is given as $\{(x_1^{(n)}, p_1^{(n)}) \cdots (x_L^{(n)}, p_L^{(n)})\}$ where, once more, each $p_i^{(n)}$ is set equal to $1/L$. In practice, the lag parameter α in this algorithm is taken to be much larger than that in algorithms of Sections 4.2, 4.3.

4.5 Regularized Particle Filters (cf. [19], [10] Chapter 12).

One common difficulty with particle filters is that they suffer from the lack of diversity among particles. This problem can be particularly severe in settings where there are degeneracies in the signal dynamics. In order to treat this difficulty one usually considers *regularized particle filters* which corresponds to replacing the sampling from the discrete distribution $\{(x_i^{(n)}, p_i^{(n)}), i = 1, \dots, L\}$ by that

from an absolutely continuous approximation. The key idea is to use kernel density smoothers. The two basic versions of such regularized particle filters are as follows.

Regularization at the updating step. A regularization kernel K is a symmetric probability function on \mathbb{R}^k satisfying $\int_{\mathbb{R}^k} xK(x)dx = 0$ and $\int_{\mathbb{R}^k} |x|^2K(x)dx < \infty$. For a smoothing parameter $\gamma \in (0, \infty)$, referred to as the *bandwidth*, denote $K_\gamma(x) \doteq \frac{1}{\gamma^k}K(x/\gamma)$. The two commonly used kernels are the Gaussian kernel and the Epanechnikov kernel (cf. [10]). The basic algorithm is as follows. We consider the regularization of the algorithm in Section 4.2 with the lag parameter $\alpha = 1$. Similar procedures can be incorporated for other particle filtering algorithms as well. Suppose that an approximation Π_{n-1}^* as a discrete distribution $\{(x_1^{(n-1)}, 1/L), \dots, (x_L^{(n-1)}, 1/L)\}$ is given. An approximation $\hat{\Pi}_n^*$ to $\hat{\Pi}_n$ as the discrete probability distribution $\{(\hat{x}_1^{(n)}, 1/L), \dots, (\hat{x}_L^{(n)}, 1/L)\}$ is obtained as in Section 4.1. Update the weights $\hat{p}_j^{(n)} \mapsto p_j^{(n)}$ using the observation Y_n by setting $p_j^{(n)} \doteq cp_j^{(n-1)}\tilde{R}(\hat{x}_j^{(n)}, Y_n)$, where c is the normalization constant. Draw $\{\tilde{x}_i^{(n)}, i = 1, \dots, L\}$ from the discrete distribution $\{(\hat{x}_1^{(n)}, p_1^{(n)}), \dots, (\hat{x}_L^{(n)}, p_L^{(n)})\}$ and generate $\{\varepsilon_i\}_{i=1}^L$, i.i.d. from the kernel K . Define $x_n^i \doteq \tilde{x}_n^i + \gamma\varepsilon_i$. The filter Π_n^* is then given as $\{(x_1^{(n)}, 1/L) \cdots (x_L^{(n)}, 1/L)\}$. As $\gamma \rightarrow 0$ and $L \rightarrow \infty$ the approximate filter Π_n^* converges to the optimal filter Π_n . Such a random jiggling of the particle locations can be particularly important if the stochastic dynamical system governing the state evolution is very sensitive to the initial condition.

Regularization at the prediction step with acceptance/rejection. A slight modification of the above regularization scheme is as follows. Here, the regularization is done at the prediction step. Once more, for the sake of definiteness consider the algorithm in Section 4.2 with $\alpha = 1$. Suppose that an approximation Π_{n-1}^* as a discrete distribution $\{(x_1^{(n-1)}, 1/L) \cdots (x_L^{(n-1)}, 1/L)\}$ is given. Then the recursion is given as follows. Obtain the approximation $\hat{\Pi}_n^*$ to $\hat{\Pi}_n$ as the discrete probability distribution $\{(\hat{x}_1^{(n)}, 1/L) \cdots (\hat{x}_L^{(n)}, 1/L)\}$ just as in Section 4.1. Set $p_j^{(n)} \doteq c\tilde{R}(\hat{x}_j^{(n)}, Y_n)$, where c is the normalization constant.

Acceptance/Rejection step. Generate a particle \tilde{x} from the discrete distribution $\{(\hat{x}_1^{(n)}, p_1^{(n)}), \dots, (\hat{x}_L^{(n)}, p_L^{(n)})\}$ and ε from the kernel K and U from a Uniform distribution on $[0, 1]$. Let $x = \tilde{x} + \gamma\varepsilon$. If $\tilde{R}(x, Y_n) > U \sup_{z \in \mathbb{R}^k} \tilde{R}(z, Y_n)$, accept x as a particle for the n th time step. Otherwise, reject the particle.

Repeat the acceptance/rejection step until L particles : $\{x_i^n\}_{i=1}^L$ are obtained. The filter Π_n^* is then given as $\{(x_1^{(n)}, 1/L) \cdots (x_L^{(n)}, 1/L)\}$. Once again, it can be shown that, as $\gamma \rightarrow 0$ and $L \rightarrow \infty$ the approximate filter Π_n^* converges to the optimal filter Π_n .

5 An example.

In this section we present an example to illustrate that in settings where the signal-observation dynamics are nonlinear, an approximation to the nonlinear filter that is not based on linearization of the model can significantly outperform the EKF. The dynamics that are described below in equation (5.11) model the motion of ship which moves with a constant (positive) radial and angular

velocity, perturbed by a white noise, when it is “not too far” away from a chosen center; whereas when the ship drifts too far off from the center a gentle restoring force pushes it back towards the center. The state process $X(t) = (X_1(t), X_2(t))'$ is described by the following SDEs

$$\begin{aligned} dX_1(t) &= -X_2(t)dt + f_1(X_1(t), X_2(t))dt + e_1dW_1(t), \\ dX_2(t) &= X_1(t)dt + f_2(X_1(t), X_2(t))dt + e_2dW_2(t), \end{aligned} \quad (5.11)$$

where W_1, W_2 are independent Brownian motions, e_1, e_2 are positive constants and

$$\begin{aligned} f_1(x_1, x_2) &\doteq \gamma \frac{x_1}{x_1^2 + x_2^2} - K \frac{x_1}{\sqrt{x_1^2 + x_2^2}} 1_{\sqrt{x_1^2 + x_2^2} \geq R}(x), \\ f_2(x_1, x_2) &\doteq \gamma \frac{x_2}{x_1^2 + x_2^2} - K \frac{x_2}{\sqrt{x_1^2 + x_2^2}} 1_{\sqrt{x_1^2 + x_2^2} \geq R}(x). \end{aligned}$$

A somewhat similar model, described via a four dimensional stochastic dynamical system, was studied in [5]. A simulation of the signal trajectory, for the parameter values $\gamma = 2, K = 50, R = 9, X(0) = (0.5, -0.5)'$, over the time interval $[0, 12.5]$, using a forward Euler’s scheme for the SDE (5.11) is shown in Figure 1. The observations are collected at intervals of length $\Delta = 0.05$ and are

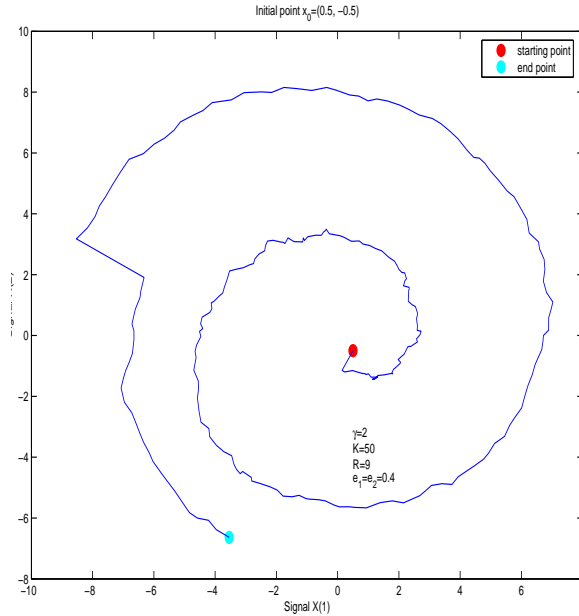


Figure 1: A Typical Signal Trajectory

given via an angular measurement, corrupted by noise. More precisely,

$$Y_n = g^*(X_1(n\Delta), X_2(n\Delta)) + \theta\xi_n, \quad (5.12)$$

where ξ_n are i.i.d. $N(0, 1)$ independent of $(X(0), W)$ and $g^*(x_1, x_2) \doteq \arctan(x_2/x_1)$. For our numerical study we chose $\theta = 0.32$. Figure 2 illustrates the difficulty with the Extended Kalman

Filter. It is seen to initially perform quite well (in fact better than the particle filter) but it soon loses the track and, subsequently, the output becomes essentially meaningless. The particle filter (we chose the particle filter of Section 4.2 with lag parameter $\alpha = 1$ and $L = 500$) on the other hand needs some time before the track is established but thereafter it tracks the signal consistently. In order to do a detailed analysis we simulated 100 signal-observation trajectories of time length

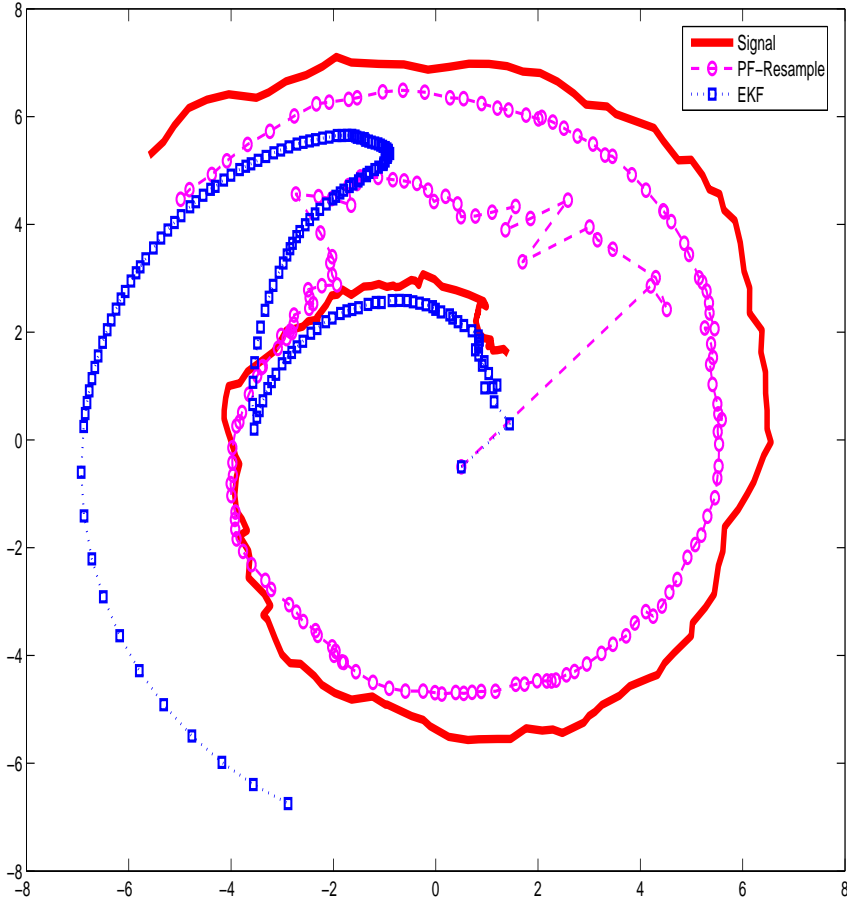


Figure 2: A comparison of EKF with a particle filter

8.25 and implemented the EKF and particle filters described in Sections 4.1, 4.2, 4.3. The number of particles for each of the particle filters was fixed to be 500. For the EKF, $m(0), P(0)$ were taken to be $(0.5, -0.5)'$ and $10\mathcal{I}_{2 \times 2}$ respectively, where $\mathcal{I}_{2 \times 2}$ is a 2×2 identity matrix. For the particle filter of Section 4.2 we considered two cases: One where the lag parameter is set to 1 and another with the lag parameter 2. These filters are referred to in Table 1 as *PF_Resample* and *PF_Resample.lag*. For the filter in Section 4.3 we chose the lag parameter to be 2. In Table 1 this filter is referred to as *PF_var_reduction*. Finally, EKF refers to the extended Kalman filter

and *PF_simple* to the particle filter of Section 4.1. As a measure for comparing performances we computed the root mean square (rms) error, d over all trajectories and over all time instants. More precisely, we computed for each filter

$$d^2 = \frac{1}{100} \frac{1}{165} \sum_{j=1}^{100} \sum_{i=1}^{165} |X^j(i\Delta) - \hat{X}^j(i\Delta)|^2$$

where $X^j, j = 1, \dots, 100$ represents the j -th simulation of the signal trajectory, $\hat{X}^j(i\Delta)$ the state estimate obtained for the time instant $i\Delta$ from the corresponding filter and $165 = 8.25/.05$ is the total number of observation time steps. As another measure we computed the proportion of runs where the filter lost the track. By “losing the track” we mean that the distance between the state estimate and actual state value at some time instant becomes larger than some pre-specified tolerance level. Here, the tolerance level of the squared distance was chosen to be 22. Table 1 summarizes our results. It is seen that the rms error for the EKF is almost six times as large as that for the best particle filter. Although all the particle filters are seen to exhibit quite similar behavior, it is seen that resampling improves the estimation procedure. The scheme where resampling is done at a lag of 2 using a variance reduction scheme is seen to perform the best. In regards to our second measure of performance, we see that Kalman filter loses the track in 99 out of 100 runs. The simplest form of the particle filter (Section 4.1) doesn’t do much better. This poor performance can be attributed to the severe degeneracies associated with particle weights which in turn results in very unstable estimated tracks. Resampling is seen to drastically improve the success rate. In particular, for the best scheme (variance reduction scheme with lag 2) only one out of 100 runs resulted in a lost track. Although this is quite a simple example, it illustrates very well the dramatic improvements that one can achieve over an EKF by using a suitable form of a nonlinear scheme for filter approximation.

Type of filter	rms error d	rate of success
EKF	10.6922	0.0100
PF_Simple	2.6080	0.0900
PF_Resample	1.9121	0.9700
PF_Resample_lag	1.7856	0.9500
PF_var_reduction	1.6954	0.9900

Table 1: Performance comparison

References

- [1] V.E .Beneš, Exact finite-dimensional filters for certain diffusions with nonlinear drift. *Stochastics*, 5:65–92, 1981.
- [2] A. Bensoussan, R. Glowinski and A. Rascanu, Approximations of Zakai equation by the splitting up method. In Zabczyk, editor. *Stochastic Systems and Optimization*, 257–265, 1989.

- [3] A. Budhiraja and G. Kallianpur, Approximation to solutions of Zakai equations using multiple Wiener and Stratonovich expansions. *Stochastics*, 56: 271–315, 1996.
- [4] A. Budhiraja and G. Kallianpur, The Feynman-Stratonovich semigroup and Stratonovich integral expansions in nonlinear filtering. *App. Math. Opt.*, 35: 177–202, 1997.
- [5] A. Budhiraja and H. J. Kushner, A nonlinear filtering algorithm based on an approximation of the conditional distribution. *IEEE Trans. Aut. Contr.*, 45: 580–585, 2000.
- [6] A. Budhiraja and H. J. Kushner, Approximation and limit results for nonlinear filters over an infinite time interval- II. Random sampling algorithms. *SIAM J. Control Optim.*, 38: 1874–1908, 2000.
- [7] J. Carpenter, P. Clifford and P. Fearnhead, An improved particle filter for nonlinear problems. *IEE Proceedings-Radar Sonar and Navigation*, 146:2–7.
- [8] D. Crisan, P. Del Moral and T. Lyons, Discrete filtering using branching and interacting particle systems. *Markov Processes and Related Fields*, 5: 293–318.
- [9] D. Crisan and T. Lyons, Nonlinear filtering and measure valued processes. *Probability Theory and Related Fields*, 109: 217–244.
- [10] A. Doucet, N. de Freitas, N. Gordon, eds. Sequential Monte Carlo Methods in Practice. *Statistics for Engineering and Information Science*.
- [11] N. J. Gordon, D. J. Salmond and A. F. .M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE-Proceedings-F*, 140:107–113.
- [12] A. H. Jazwinski *Stochastic Processes and Filtering*. Academic Press, 1970.
- [13] G. Kallianpur. *Stochastic Filtering Theory*. Springer-Verlag, 1980.
- [14] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME Ser. D J. Basic Eng.*, 83:95–108, 1961.
- [15] G. Kitagawa. Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82:1032–1063, 1987.
- [16] H. J. Kushner. Approximations to optimal nonlinear filters. *IEEE Trans. Automat. Contr.*, 12:546-556, 1967.
- [17] H. J. Kushner. *Probability methods for approximations in stochastic control and for elliptic equations*. Academic Press, 1977.
- [18] H. J. Kushner and P. Dupuis. *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer-Verlag, New York, second edition, 2001.
- [19] F. Le Gland, C. Musso and N. Oudjane. An analysis of regularized interacting particle methods for nonlinear filtering. *Proceedings of the IEEE European workshop on computer-intensive methods in control and data processing, Prague*. pp. 167-174, 1998.

- [20] R. S. Liptser and A. N. Shiryaev *Statistics of random processes II*. Springer-Verlag, 1978.
- [21] S. Lototsky, R. Mikulevicius and B. Rozovskii. Nonlinear filtering revisited: a spectral approach. *SIAM J. Control Optim.*, 35:435–461, 1997.