Implicit sampling for particle filters

Alexandre J. Chorin * and **Xuemin Tu** *

* Department of Mathematics, University of California at Berkeley and Lawrence Berkeley National Laboratory Berkeley, CA 94720

We present a particle-based nonlinear filtering scheme, related to recent work on chainless Monte Carlo, designed to focus particle paths sharply so that fewer particles are required. The main features of the scheme are a representation of each new probability density function by means of a set of functions of Gaussian variables (a distinct function for each particle and step) and a resampling based on normalization factors and Jacobians. The construction is demonstrated on a standard ill-conditioned test problem.

Implicit sampling | filter | pseudo-Gaussian | Jacobian | chainles sampling

Introduction.

There are many problems in science in which the state of a system must be identified from an uncertain equation supplemented by a stream of noisy data (see e.g. [1]). A natural model of this situation consists of a stochastic differential equation (SDE):

$$dx = f(x, t)\,dt + g(x, t)\,dw, \qquad [1]$$

where $x = (x_1, x_2, \ldots, x_m)$ is an m-dimensional vector, w is m-dimensional Brownian motion, f is an m-dimensional vector function, and $g(x, t)$ is an m by m diagonal matrix. The Brownian motion encapsulates all the uncertainty in this equation. The initial state $x(0)$ is assumed given and may be random as well.

As the experiment unfolds, it is observed, and the values b^n of a measurement process are recorded at times t^n. For simplicity assume $t^n = n\delta$, where δ is a fixed time interval and n is an integer. The measurements are related to the evolving state $x(t)$ by

$$b^n = h(x^n) + GW^n, \qquad [2]$$

where h is a k-dimensional, generally nonlinear, vector function with $k \leq m$, G is a diagonal matrix, $x^n = x(n\delta)$, and W^n is a vector whose components are independent Gaussian variables of mean 0 and variance 1, independent also of the Brownian motion in equation (1). The task is to estimate x on the basis of equation (1) and the observations (2).

If the system (1) and equation (2) are linear and the data are Gaussian, the solution can be found via the Kalman-Bucy filter (see e.g. [2]). In the general case, it is natural to try to estimate x via its evolving probability density. The initial state x is known and so is its probability density; all one has to do is evaluate sequentially the density P_{n+1} of x^{n+1} given the probability densities P_k of x^k for $k \leq n$ and the data b^{n+1}. This can be done by following "particles" (replicas of the system) whose empirical distribution approximates P_n. In a Bayesian filter (see e.g [3, 4, 5, 6, 7, 8, 9, 10]), one uses the pdf P_n and equation (1) to generate a prior density, and then one uses the new data b^{n+1} to generate a posterior density P_{n+1}. In addition, one has to sample backward to take into account the information each measurement provides about the past, as well as avoid having too many identical particles. This can be very expensive, in particular because the number of particles needed can grow catastrophically (see e.g. [11, 12]).

In this paper we offer an alternative to the standard approach, in which P_{n+1} is sampled more directly and backward sampling is done without chains [13]. Our direct sampling is based on a pseudo-Gaussian representation of a variable with density P_{n+1}, i.e. a representation by a collection of functions of Gaussian variables with sample-dependent parameters. The construction is related to chainless sampling as described in [13]. The idea in chainless sampling is to produce a sample of a large set of variables by sequentially sampling a growing sequence of nested conditionally independent subsets, with discrepancies balanced by sampling weights. As observed in [14, 15], chainless sampling for a SDE reduces to interpolatory sampling, as explained below. Our construction will be explained in the following sections through an example where the position of a ship is deduced from the measurements of an azimuth, already used as a test bed in a number of previous papers (see e.g. [7, 16, 17]). We call our sampling "implicit" by analogy with implicit schemes for solving differential equations, where the determination of a next value requires the solution of algebraic equations.

If the SDE (1) and observation equation (2) are linear, our construction becomes a reformulation of sequential importance sampling with an optimal importance function, see [5, 6].

Sampling by interpolation and iteration.

First we explain how to sample via interpolation and iteration in a simple problem, related to the example and the construction in [14]. Consider the scalar SDE

$$dx = f(x, t)dt + \sqrt{\beta}\,dw, \qquad [3]$$

where β is a constant. We want to find sample paths $x = x(t)$, $0 \leq t \leq 1$, subject to the conditions $x(0) = 0$, $x(1) = X$.

Let $N(a, v)$ denote a Gaussian variable with mean a and variance v. We first discretize equation (3) on a regular mesh $t^0, t^1, \ldots, t^N$, where $t^n = n\delta$, $\delta = 1/N$, $0 \leq n \leq N$, with $x^n = x(t^n)$, and, following [14], use a balanced implicit discretization [18, 19]:

$$x^{n+1} = x^n + f(x^n, t^n)\delta + (x^{n+1} - x^n)f'(x^n, t^n)\delta + W^{n+1},$$

where $f'(x^n, t^n) = \frac{\partial f}{\partial x^n}(x^n, t^n)$ and W^{n+1} is $N(0, \beta/N)$. The joint probability density of the variables $x^1, \ldots, x^{N-1}$ is $Z^{-1}\exp(-\sum_0^{N-1} V_n)$, where Z is the normalization constant and

$$V_n = \frac{\left((1 - \delta f')(x^{n+1} - x^n) - \delta f\right)^2}{2\beta\delta}$$

$$= \frac{\left(x^{n+1} - x^n - \delta f/(1 - \delta f')\right)^2}{2\beta_n},$$

where f, f' are functions of the x^n, t^n, and $\beta_n = \beta\delta/(1 - \delta f')^2$ (see [20]). One can obtain sample solutions by sampling this density, e.g. by Markov chain Monte Carlo, or one can obtain them by interpolation (chainless sampling), as follows.

Reserved for Publication Footnotes

Let $a_n = f(x^n, t^n)\delta/(1 - \delta f'(x^n, t^n))$. Consider first the special case $f(x, t) = f(t)$, so that in particular $f' = 0$; we recover a version of a Brownian bridge (see [21]). Each increment $x^{n+1} - x^n$ is now a $N(a_n, \beta/N)$ variable, with the $a_n = f(t^n)\delta$ known explicitly. Let N be a power of 2. Consider the variable $x^{N/2}$. On one hand,

$$x^{N/2} = \sum_1^{N/2} (x^n - x^{n-1}) \sim N(A_1, V_1),$$

where $A_1 = \sum_1^{N/2} a_n$, $V_1 = \beta/2$. On the other hand,

$$X = x^{N/2} + \sum_{N/2+1}^{N} (x^n - x^{n-1}),$$

so that

$$x^{N/2} \sim N(A_2, V_2),$$

with

$$A_2 = X - \sum_{N/2}^{N-1} a_n, \quad V_2 = V_1.$$

The pdf of $x^{N/2}$ is the product of the two pdfs; one can check that

$$\exp\left(-\frac{(x - A_1)^2}{2V_1}\right) \exp\left(-\frac{(x - A_2)^2}{2V_2}\right)$$
$$= \exp\left(-\frac{(x - \bar{a})^2}{2\bar{v}}\right) \exp(-\phi),$$

where $\bar{v} = \frac{V_1 V_2}{V_1 + V_2}$, $\bar{a} = \frac{V_2 A_1 + V_1 A_2}{V_1 + V_2}$, and $\phi = \frac{(A_2 - A_1)^2}{2(V_1 + V_2)}$; $e^{-\phi}$ is the probability of getting from the origin to X, up to a normalization constant.

Pick a sample ξ_1 from the $N(0, 1)$ density; one obtains a sample of $x^{N/2}$ by setting $x^{N/2} = \bar{a} + \sqrt{\bar{v}}\xi_1$. Given a sample of $x^{N/2}$ one can similarly sample $x^{N/4}$, $x^{3N/4}$, then $x^{N/8}$, $x^{3N/8}$, etc., until all the x^j have been sampled. If we define $\xi = (\xi_1, \xi_2, \ldots, \xi_{N-1})$, then for each choice of ξ we find a sample $\mathbf{x} = (x^1, \ldots, x^{N-1})$ such that

$$\exp\left(-\frac{\xi_1^2 + \cdots + \xi_{N-1}^2}{2}\right) \exp\left(-\frac{(X - \sum_n a_n)^2}{2\beta}\right)$$
$$= \exp\left(-\frac{(x^1 - x^0 - a_0)^2}{2\beta/N} - \frac{(x^2 - x^1 - a_1)^2}{2\beta/N}\right.$$
$$\left. - \cdots - \frac{(x^N - x^{N-1} - a_{N-1})^2}{2\beta/N}\right), \qquad \textbf{[4]}$$

where the factor $\exp\left(-\frac{(X - \sum_n a_n)^2}{2\beta}\right)$ on the left is the probability of the fixed end value X up to a normalization constant. In this linear problem, this factor is the same for all the samples and therefore harmless. The Jacobian J of the variables $x^1, \ldots, x^{N-1}$ with respect to the variables $\xi^1, \ldots, \xi^{N-1}$ can be seen to be a constant independent of the sample and is also immaterial. One can repeat this sampling process for multiple choices of the variables ξ; each sample of the corresponding set of x^n is independent of any previous samples of this set.

Now return to the general case. The functions f, f' are now functions of the x^j. We obtain a sample of the probability density we want by iteration. The simplest iteration proceeds as follows. First pick $\xi = (\xi_1, \xi_2, \ldots, \xi_{N-1})$, where each $\xi_l, l = 1, \ldots, N-1$, is drawn independently from the $N(0, 1)$ density (this vector remains fixed during the iteration). Make a first guess $\mathbf{x}_0 = (x_0^1, x_0^2, \ldots, x_0^{N-1})$ (for example, if $X \neq 0$, pick $\mathbf{x}_0 = 0$). Evaluate the functions f, f' at $\mathbf{x}_j$ (note that now $f' \neq 0$, and therefore the variances of the various displacements are no longer constants). We are back in the previous case,

and can find values of the increments $x_{j+1}^{n+1} - x_{j+1}^n$ corresponding to the values of f, f' we have. Repeat the process starting with the new iterate. If the vectors $\mathbf{x}_j$ converge to a vector $\mathbf{x} = (x^1, \ldots, x^{N-1})$, we obtain, in the limit, equation (4), where now on the right side β depends on n so that $\beta = \beta_n$, and both a_n, β_n are functions of the final $\mathbf{x}$. The left hand side of (4) becomes:

$$\exp\left(-\frac{\xi_1^2 + \cdots + \xi_{N-1}^2}{2}\right) \exp\left(-\frac{(X - \sum_n a_n)^2}{2\sum_n \beta_n}\right).$$

The factor $F = \exp\left(-\frac{(X - \sum_n a_n)^2}{2\sum_n \beta_n}\right)$ is now different from sample to sample, and changes the relative weights of the different samples. The Jacobian J of the x variables with respect to the ξ variables is now also a function of the sample. It can be evaluated step by step the last time the iteration is carried out, either by an implicit differentiation, or by repeating the iteration for a slightly different value of the relevant ξ and differencing. In averaging, one should take the product $F|J|$ as weight, or resample as described at the end of the following section. In order to obtain more uniform weights, one also can use the strategies in [13, 14].

One can readily see that this iteration converges if

$$KL < 1, \qquad \textbf{[5]}$$

where K is the Lipshitz constant of f, and L is the length of the interval on which one works (here $L = 1$). If this iteration fails to converge, more sophisticated iterations are available. One should of course choose N large enough so that the results are converged in N. We do not provide more details here because they are extraneous to our purpose, which is to explain chainless/interpolatory sampling and the use of reference variables in a simple context.

Finally, we chose the reference density to be a product of independent $N(0, 1)$ variables, which is a convenient but not mandatory choice. In applications one may well want to choose other variances or make the variables be dependent.

The ship azimuth problem.

The problem we focus on is discussed in [7, 16, 17], where it is used to demonstrate the capabilities of particular filters. A ship sets out from a point (x^0, y^0) in the plane and undergoes a random walk,

$$x^{n+1} = x^n + u^{n+1},$$
$$y^{n+1} = y^n + v^{n+1}, \qquad \textbf{[6]}$$

for $n \geq 0$, $u^{n+1} = N(u^n, \beta)$, $v^{n+1} = N(v^n, \beta)$, i.e., each displacement is a sample of a Gaussian random variable whose variance β does not change from step to step and whose mean is the value of the previous displacement. An observer makes noisy measurements of the azimuth $\arctan(y^n/x^n)$ (for the sake of definiteness, we take the branch in $-[\pi/2, \pi/2)$), recording

$$b^n = \arctan\frac{y^n}{x^n} + N(0, s). \qquad \textbf{[7]}$$

where the variance s is also fixed; here the observed quantity b^n is scalar and is not be denoted by a boldfaced letter. The problem is to reconstruct the positions $\mathbf{x}^n = (x^n, y^n)$ from equations (6,7). We take the same parameters as [7]: $x^0 = 0.01, y^0 = 20, u^1 = 0.002$, $v^1 = -0.06, \beta = 1 \cdot 10^{-6}, s = 25 \cdot 10^{-6}$. We follow numerically M particles, all starting from $X_i^0 = x^0, Y_i^0 = y^0$, as described in the following sections, and we estimate the ship's position at time $n\delta$ as the mean of the locations $\mathbf{X}_i^n = (X_i^n, Y_i^n), i = 1, \ldots, M$ of the particles at that time. The authors of [7] also show numerical results for runs with varying data and constants; we discuss those refinements in the numerical results section below.

Forward step.

Assume we have a collection of M particles $\mathbf{X}^n$ at time $t^n = n\delta$ whose empirical density approximates P_n; now we find displacements $\mathbf{U}^{n+1} = (U^{n+1}, v^{n+1})$ such that the empirical density of $\mathbf{X}^{n+1} = \mathbf{X}^n + \mathbf{U}^{n+1}$ approximates P_{n+1}. P_{n+1} is known implicitly: it is the product of the density that can be deduced from the dynamics and the one that comes from the observations, with the appropriate normalization. If one is given sample displacements, their probabilities p (the densities P_{n+1} evaluated at the resulting positions $\mathbf{X}^{n+1}$) can be evaluated, so p is a function of $\mathbf{U}^{n+1}$, $p = p(\mathbf{U}^{n+1})$. For each particle i, we are going to sample a reference density, obtain a reference sample of probability ρ, then attempt to solve (by iteration) the equation

$$\rho = p(\mathbf{U}_i^{n+1}) \qquad [8]$$

to obtain $\mathbf{U}_i^{n+1}$.

Define $f(x, y) = \arctan(y/x)$ and $f^n = f(X^n, Y^n)$. We are working on one particle at a time, so the index i can be temporarily suppressed. Pick two independent samples ξ_x, ξ_y from a $N(0, 1)$ density (the reference density in the present calculation), and set $\rho = \frac{1}{2\pi} \exp\left(-\frac{\xi_x^2}{2} - \frac{\xi_y^2}{2}\right)$; the variables ξ_x, ξ_y remain unchanged until the end of the iteration. We are looking for displacements U^{n+1}, V^{n+1}, and parameters a_x, a_y, v_x, v_y, ϕ, such that:

$$2\pi\rho = \exp\left(-\frac{(U^{n+1} - U^n)^2}{2\beta} - \frac{(V^{n+1} - V^n)^2}{2\beta}\right.$$
$$\left. -\frac{(f^{n+1} - b^{n+1})^2}{2s}\right) \exp(\phi)$$
$$= \exp\left(-\frac{(U^{n+1} - a_x)^2}{2v_x} - \frac{(V^{n+1} - a_y)^2}{2v_y}\right) \qquad [9]$$

The first equality states what we wish to accomplish. Indeed, divide this first equality by $\exp(\phi)$. The equality now defines implicitly new displacements U^{n+1}, V^{n+1}, functions of ξ_x, ξ_y, with the probability of these displacements with respect to P_{n+1} given (up to an unknown normalization constant). The second equality in equations (9) defines parameters a_x, a_y, v_x, v_y, (all functions of $\mathbf{X}^n$ and ξ_x, ξ_y) that will be used to actually find the displacements U^{n+1}, V^{n+1}. One should remember that in our example the mean of U^{n+1} before the observation is taken into account is U^n, and similarly for V^{n+1}.

We use the second equality in (9) to set up an iteration for vectors $\mathbf{U}^{n+1,j}$ ($= \mathbf{U}^j$ for brevity) that converges to $\mathbf{U}^{n+1}$. Start with $\mathbf{U}^0 = 0$. We now explain how to compute $\mathbf{U}^{j+1}$ given $\mathbf{U}^j$.

Approximate the observation equation (7) by

$$f(\mathbf{X}^j) + f_x \cdot (U^{j+1} - U^j) + f_y \cdot (V^{j+1} - V^j) = b^{n+1} + N(0, s), \qquad [10]$$

where the derivatives f_x, f_y are, like f, evaluated at $\mathbf{X}^j = \mathbf{X}^n + \mathbf{U}^j$, i.e., approximate the observation equation by its Taylor series expansion around the previous iterate. Define a variable $\eta^{j+1} = (f_x \cdot U^{j+1} + f_y \cdot V^{j+1})/\sqrt{f_x^2 + f_y^2}$. The approximate observation equation says that η^{j+1} is a $N(a_1, v_1)$ variable, with

$$a_1 = -\frac{f - f_x \cdot U^j - f_y \cdot V^j - b^{n+1}}{\sqrt{f_x^2 + f_y^2}},$$

$$v_1 = \frac{s}{f_x^2 + f_y^2}. \qquad [11]$$

On the other hand, from the equations of motion one finds that η^{j+1} is $N(a_2, v_2)$, with $a_2 = (f_x \cdot U^n + f_y \cdot V^n)/\sqrt{f_x^2 + f_y^2}$ and $v_2 = \beta$. Hence the pdf of η^{j+1} is, up to normalization factors,

$$\exp\left(-\frac{(x - a_1)^2}{2v_1} - \frac{(x - a_2)^2}{2v_2}\right) = \exp\left(-\frac{(x - \bar{a})^2}{2\bar{v}}\right) \exp(-\phi),$$

where $\bar{v} = \frac{v_1 v_2}{v_1 + v_2}, \bar{a} = \frac{a_1 v_2 + a_1 v_2}{v_1 + v_2}, \phi = \frac{(a_1 - a_2)^2}{2(v_1 + v_2)} = \phi^{j+1}$.

We can also define a variable η_+^{j+1} that is a linear combination of U^{j+1}, V^{j+1} and is uncorrelated with η^{j+1}:

$$\eta_+^{j+1} = \frac{-f_y \cdot U^{j+1} + f_x \cdot V^{j+1}}{\sqrt{f_x^2 + f_y^2}}.$$

The observations do not affect η_+^{j+1}, so its mean and variance are known. Given the means and variances of η^{j+1}, η_+^{j+1} one can easily invert the orthogonal matrix that connects them to U^{j+1}, V^{j+1} and find the means and variances a_x, v_x of U^{j+1} and a_y, v_y of V^{j+1} after their modification by the observation (the subscripts on a, v are labels, not differentiations). Now one can produce values for U^{j+1}, V^{j+1}:

$$U^{j+1} = a_x + \sqrt{v_x}\xi_x, \quad V^{j+1} = a_y + \sqrt{v_y}\xi_y,$$

where ξ_x, ξ_y are the samples from $N(0, 1)$ chosen at the beginning of the iteration. This completes the iteration.

This iteration converges to $\mathbf{X}^{n+1}$ such that $f(\mathbf{X}^{n+1}) = b^{n+1} + N(0, s)$, and the phases ϕ^j converge to a limit $\phi = \phi_i$, where the particle index i has been restored. The time interval over which the solution is updated in each step is short, and there are no problems with convergence, either here or in the next section (see equation (5)); in all cases the iteration converges in a small number of steps.

We now calculate the Jacobian J of the $\mathbf{U}^{n+1}$ variables with respect to ξ_x, ξ_y. The relation between these variables is laid out in the first of equations (9). Take the log of this equation, partition it into a part parallel to the direction in which the observation is made (i.e., parallel to the vector (f_x, f_y)) and a part orthogonal to that direction. Since the increment U^{n+1}, V^{n+1} is now known, the evaluation of J is merely an exercise in implicit differentiation. J can also be evaluated numerically, by finding the increment U^{n+1}, V^{n+1} that corresponds to nearby values of ξ_x, ξ_y, and differencing.

Do this for all the particles and obtain new positions with weights $W_j = \exp(-\phi_j)|J_j|$, where J_j is the Jacobian for the j-th particle. One can get rid of the weights by resampling, i.e., for each of M random numbers $\theta_k, k = 1, \ldots, M$ drawn from the uniform distribution on $[0, 1]$, choose a new $\widehat{\mathbf{X}}_k^{n+1} = \mathbf{X}_i^{n+1}$ such that $A^{-1} \sum_{j=1}^{i-1} W_j < \theta_k \leq A^{-1} \sum_{j=1}^{i} W_j$ (where $A = \sum_{j=1}^{M} W_j$), and then suppress the hat. We have traded the usual Bayesian resampling based on the posterior probabilities of the samples for a resampling based on the normalizing factors of the several Gaussian densities; this is a worthwhile trade because in a Bayesian filter one gets a set of samples many of which may have low probability with respect to P_{n+1}, and here we have a set of samples each one of which has high probability with respect to a pdf close to P_{n+1} (see the numerical results and conclusion sections).

Note also that the resampling does not have to be done at every step- for example, one can add up the phases for a given particle and resample only when the ratio of the largest cumulative weight $\exp(-\sum(\phi_i - \log|J_i|))$ to the smallest such weight exceeds some limit L (the summation is over the weights accrued to a particular particle i since the last resampling). If one is worried by too many particles being close to each other ("depletion" in the Bayesian terminology), one can divide the set of particles into subsets of small size and resample only inside those subsets, creating a greater diversity. As will be seen in the numerical results section, none of these strategies will be used here and we will resample fully at every step.

Finally, note that if the SDE (1) and observation equation (2) are linear, and if at time $n\delta$ one is given the means and the covariance matrix of a Gaussian $\mathbf{x}$, then our algorithm produces, in one iteration, the means and the covariance matrix of a standard Kalman filter.

Backward sampling.

The algorithm of the previous section is sufficient to create a filter, but accuracy, when the problem is not Gaussian, may require an additional step. Every observation provides information not only about the future but also about the past- it may, for example, tag as improbable earlier states that had seemed probable before the observation was made; in general one has to go back and correct the past after every observation (this backward sampling is often misleadingly motivated solely by the need to create greater diversity among the particles in a Bayesian filter). As will be seen below, this backward sampling does not provide a significant boost to accuracy in the present problem, but it must be described for the filter to be of general use, as well as be generalizable to problems involving smoothing.

Given a set of particles at time $(n + 1)\delta$, after a forward step and maybe a subsequent resampling, one can figure out where each particle i was in the previous two steps, and have a partial history for each particle i: $\mathbf{X}_i^{n-1}, \mathbf{X}_i^n, \mathbf{X}_i^{n+1}$ (if resamples had occurred, some parts of that history may be shared among several current particles). Knowing the first and the last members of this sequence, one can interpolate for the middle term as in section 2, thus projecting information backward. This requires that one recompute $\mathbf{U}^n$.

Let $\mathbf{U}^{\text{tot}} = \mathbf{U}^n + \mathbf{U}^{n+1}$; in the present section this quantity is assumed known and remains fixed. In the azimuth problem discussed here, one has to deal with the slight complication due to the fact that the mean of each displacement is the value of the previous one, so that two successive displacements are related in a slightly more complicated way than usual. The displacement U^n is a $N(U^{n-1}, \beta)$ variable, and U^{n+1} is a $N(U^n, \beta)$ variable, so that one goes from X^{n-1} to X^{n+1} by sampling first a $(2U^{n-1}, 4\beta)$ variable that takes us from X^{n-1} to an intermediate point P, with a correction by the observation half way up this first leg, and then one samples a $N(U^{\text{tot}}, \beta)$ variable to reach X^{n+1}, and similarly for Y. Let the variable that connects $\mathbf{X}^{n-1}$ to P be $\mathbf{U}^{\text{new}}$, so that what replaces $\mathbf{U}^n$ is $\mathbf{U}^{\text{new}}/2$. Accordingly, we are looking for a new displacement $\mathbf{U}^{\text{new}} = (U^{\text{new}}, V^{\text{new}})$, and for parameters $a_x^{\text{new}}, a_y^{\text{new}}, v_x^{\text{new}}, v_y^{\text{new}}$ such that

$$2\pi \exp\left(-\frac{\xi_x^2 + \xi_y^2}{2}\right)$$
$$= \exp\left(-\frac{(U^{\text{new}} - 2U^{n-1})^2}{8\beta} - \frac{(V^{\text{new}} - 2V^{n-1})^2}{8\beta}\right)$$
$$\times \exp\left(-\frac{(f^{\text{new}} - b^n)^2}{2s}\right)$$
$$\times \exp\left(-\frac{(U^{\text{new}} - U^{\text{tot}})^2}{2\beta} - \frac{(V^{\text{new}} - V^{\text{tot}})^2}{2\beta}\right)\exp(\phi)$$
$$= \exp\left(-\frac{(U^{\text{new}} - \bar{a}_x)^2}{2v_x^{\text{new}}} - \frac{(V^{\text{new}} - \bar{a}_y)^2}{2v_y^{\text{new}}}\right),$$

where $f^{\text{new}} = f(X^{n-1} + U^{\text{new}}/2, Y^{n-1} + V^{\text{new}}/2)$ and ξ_x, ξ_y are independent $N(0, 1)$ Gaussian variables. As in equation (9), the first equality embodies what we wish to accomplish- find displacements, functions of the reference variables, that sample the new pdf at time $n\delta$ defined by the forward motion, the constraint imposed by the observation, and by knowledge of the position at time $(n + 1)\delta t$. The second equality states that this is done by finding particle-dependent parameters for a Gaussian density.

We again find these parameters as well as the displacements by iteration. Much of the work is separate for the X and Y components of the equations of motion, so we write some of the equations for the X component only. Again set up an iteration for variables $U^{\text{new},j} = U^j$ which converge to U^{new}. Start with $U^0 = 0$. To find U^{j+1} given U^j, approximate the observation equation (7), as before, by equation (10); define again variables η^{j+1}, η_+^{j+1}, one in the direction of the approximate constraint and one orthogonal to it; in the direction of the constraint multiply the pdfs as in the previous section; construct

new means a_x^1, a_y^1 and new variances v_x^1, v_y^1 for at time n, taking into account the observation at time n, again as before. This also produces a phase $\phi = \phi_0$.

Now take into account that the location of the ship at time $n + 1$ is known; this creates a new mean $\bar{a}_x$, a new variance $\bar{v}_x$, and a new phase ϕ_x, by $\bar{v} = \frac{v_1 v_2}{v_1 + v_2}$, $\bar{a}_x = \frac{a_1 v_2 + a_2 v_1}{v_1 + v_2}$, $\phi_x = \frac{(a_1 - a_2)^2}{v_1 + v_2}$, where $a_1 = 2a^1, v_1 = 4v_x^1, a_2 = X^{\text{tot}}, v_2 = \beta$. Finally, find a new interpolated position $U^{j+1} = \frac{a_x^{\text{new}}}{2} + \sqrt{\frac{v_x^{\text{new}}}{4}}\xi_x$ (the calculation for V^{j+1} is similar, with a phase ϕ_y), and we are done. The total phase for this iteration is $\phi = \phi_0 + \phi_x + \phi_y$. As the iterates U^j converge to U^{new}, the phases converge to a limit $\phi = \phi_i$. One also has to compute Jacobians and set up a resampling. Once one has values for $\mathbf{X}^{\text{new}}$, a forward step gives corrected values of $\mathbf{X}^{n+1}$; one can use this interpolation process to correct estimates of $\mathbf{X}^k$ by subsequent observations for $k = n - 1, k = n - 2, \ldots$, as many as are useful.

Numerical results.

Before presenting examples of numerical results for the azimuth problem, we discuss the accuracy one can expect. We run the ship once and record synthetic observations (with the appropriate noise) which will remain fixed. Then we find other ship trajectories compatible with these fixed observations as follows. We have 160 observations. We note that the maximum likelihood estimate of s given 160 observations is a random variable with mean zero and variance $.11s$. Then we make other runs of the ship, record the azimuths along the path and calculate the differences between these azimuths and the fixed observations. If the set of these differences in any run is a likely set of 160 samples of a $N(0, s)$ variable (which what the noise is supposed to be), then we declare that the new run is compatible with the fixed observations. We view the set of differences as likely if their empirical variance is within one standard variation $(.11s)$ of the nominal variance s of the observation noise. One can impose further requirements (for example, one may demand that the empirical covariance of two successive noises be within a standard deviation of zero), but these turn out to be weaker requirements. To lighten the burden of computation we make the new ship runs have fixed displacements in the observed direction (equal to those that the first ship experienced) and sample new displacements only in the direction orthogonal to the observed direction. We use the variability of the compatible runs as a an estimate of the lower bound on the possible accuracy of the reconstruction.

In Table I we display the standard deviations of the differences between the resulting paths and the original path that produced the observations after the number of steps indicated there (the means of these differences are statistically indistinguishable from zero). This Table provides an estimate of the accuracy we can expect. It is fair to assume that these standard deviations are underestimates of the uncertainty- a maximum variation of a single standard deviation in s is a strict requirement, and we allowed no variability in β. In particular, our construction, together with the particular set of directions in the linearized observation equations that arises with our data, conspire to make the error estimate in the x-component unrealistically small.

Table I
Intrinsic uncertainty in the azimuth problem

step	x component	y component
40	.0005	.21
80	.004	.58
120	.010	.88
160	.017	.95

If one wants reliable information about the performance of the filter, it is not sufficient to run the ship once, record observations, and then use the filter to reconstruct the ship's path, because the differ-

ence between the true path and the reconstruction is a random variable which may be accidentally atypically small or atypically large. We have therefore run a large number of such reconstructions and computed the means and standard deviations of the discrepancies between path and reconstruction as a function of the number of steps and of other parameters. In Tables II and III we display the means and standard deviations of these discrepancies (not of their mean!) in the the x and y components of the paths with 2000 runs, at the steps and numbers of particles indicated, with no backward sampling. (Ref. [7] used 100 particles). On the average the error is zero so that the filter is unbiased, and the standard deviation of the discrepancies cannot be expected to be better than the lower bound of Table I, and in fact it is compatible to that lower bound. The standard deviation of the discrepancy is not catastrophically larger with one particle (and no resampling at all !) than with 100- the main source of the discrepancy is the insufficiency of the data for accurate estimation of the trajectories. The more sophisticated resampling strategies discussed above make no discernible difference here, because they are unable to remedy the limitations of the data set. One can check that backward sampling does not make much difference either for this problem where the underlying motion is Gaussian and the variance of the observation noise is much larger than the variance in the model.

Table IIa
Mean and standard variation of the discrepancy between synthetic data and their reconstruction, 2000 runs, no back step, 100 particles

n. of steps	x component		y component	
	mean	s.d.	mean	s.d.
40	.0004	.04	.001	.17
80	-.001	.04	-.01	.54
120	-.0008	.07	-.03	1.02
160	-.002	.18	-.05	1.62

Table IIb
Mean and standard variation of the discrepancy between synthetic data and their reconstruction, 2000 runs, no back step, one particle

n. of steps	x component		y component	
	mean	s.d.	mean	s.d.
40	-.003	.20	-.0008	.19
80	-.01	.54	-.019	.56
120	-.01	.84	-.027	1.04
160	-.016	.94	-.02	1.62

Figure 1: Some ship trajectories (explained in the text)

Legend:
- (1) ship path
- (2) its reconstruction
- (3) its reconstruction (i)
- (4) its reconstruction (ii)
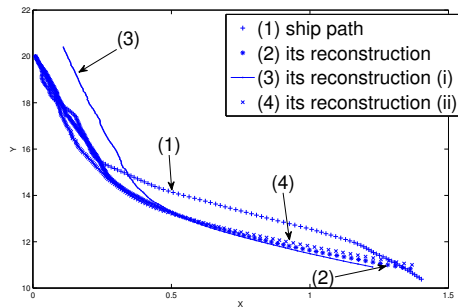
In Figure 1 we plot a sample ship path, its reconstruction, and the reconstructions obtained (i) when the initial data for the reconstruction are strongly perturbed (here, the initial data for x, y were perturbed initially by, respectively, .1 and .4), and (ii) when the value of β assumed in the reconstruction is random: $\beta = N(\beta_0, \epsilon\beta_0)$, where β_0 is the constant value used until now and $\epsilon = 0.4$ but the calculation is otherwise identical. This produces variations in β of the order of 40%; any larger variance in the perturbations produced here a negative

value of β. The differences between the reconstructions and the true path remain within the acceptable range of errors. These graphs show that the filter has little sensitivity to perturbations (we did not calculate statistics here because the insensitivity holds for each individual run).

We now show that the parameter β can be estimated from the data. The filter needs an estimate of β to function, call this estimate β_{assumed}. If $\beta_{\text{assumed}} \neq \beta$, the other assumptions used to produce the data set (e.g. independence of the displacements and of the observations) are also false, and all one has to do is detect the fallacy. We do it by picking a trajectory of a particle and computing the quantity

$$D = \frac{(\sum_2^K (U^{j+1} - dU^j))^2 + (\sum_2^K (V^{j+1} - V^j))^2}{\sum_2^K (U^{j+1} - U^j)^2 + \sum_2^K (V^{j+1} - V^j)^2}.$$

If the displacements are independent then on the average $D = 1$; we will try to find the real β by finding a value of β_{assumed} for which this happens. We chose $K = 40$ (the early part of a trajectory is less noisy than the later parts).

As we already know, a single run cannot provide an accurate estimate of β, and accuracy in the reconstruction depends on how many runs are used. In Table III we display some values of D averaged over 200 and over 3000 runs as a function of the ratio of β_{assumed} to the value of β used to generate the data. From the longer computation one can find the correct value of β with an error of about 3%, while with 200 runs the uncertainty is about 10%. The limited accuracy reported in previous work can of course be achieved with a single run. A detailed discussion of parameter estimation using our algorithm will be presented elsewhere.

Table III
The mean of the discriminant D as a function of $\sigma_{\text{assumed}}/\sigma$, 30 particles

$\sigma_{\text{assumed}}/\sigma$	3000 runs	200 runs
.5	1.15 ± .01	1.15 ± .06
.6	1.07 ± .01	1.07 ± .06
.7	1.07 ± .01	1.07 ± .05
.8	1.04 ± .01	1.04 ± .05
.9	1.02 ± .01	1.02 ± .05
1.0	1.01 ± .01	1.00 ± .05
1.1	.95 ± .01	1.01 ± .05
1.2	.95 ± .01	.95 ± .04
1.3	.94 ± .01	.96 ± .05
1.4	.90 ± .01	.88 ± .04
1.5	.89 ± .01	.88 ± .04
2.0	.85 ± .01	.83 ± .04

Conclusions.

The numerical results for the test problem are comparable to those produced by other filters. What should be noted is that our filter behaves well as the number of particles decreases (in the test problem, down to a single particle). There is no linearization or other uncontrollable approximation. This good behavior persists as the number of variables increases. The difficulty encountered by Bayesian particle filters when the number of variables increase is due to the fact that the relative size of that part of space that the data designate as probable decreases, so that it is harder for a Bayesian filter to produce probable samples. This situation can be modeled as the limit of a problem where both the variance β of the model and the variance s of the observation noise tend to zero; it is easy to see that in this limit our iteration produces the correct trajectories without difficulty. In [11, 12] Snyder, Bickel, et al. produced a simple many-dimensional problem where a Bayesian filter collapses because a single particle hogs all the probability; one can see that in that problem our filter produces the same weights for all the particles with any number of variables. We shall exhibit examples of many-dimensional nonlinear applications of our filter in subsequent publications.

1. Doucet, A., de Freitas, N., and Gordon, N. (eds) (2001), Sequential Monte Carlo Methods in Practice, Springer, New York.
2. Bozic, S. (1994)), Digital and Kalman Filtering. Butterworth-Heinemann.
3. Maceachern, S., Clyde, M., and Liu, J. (1999), Sequential importance sampling for nonparametric Bayes models: the next generation, *Can. J. Stat.* 27:251–267.
4. Liu, J. and Sabatti, C. (2000), Generalized Gibbs sampler and multigrid Monte Carlo for Bayesian computation, *Biometrika* 87:353–369.
5. Doucet, A., Godsill, S., and Andrieu, C. (2000), On sequential Monte Carlo sampling methods for Bayesian filtering, *Stat. Comp.* 10:197–208.
6. Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T. (2002), A tutorial on particle filters for online nonlinear/nongaussian Bayesian tracking, *IEEE Trans. Sig. Proc.* 50:174–188.
7. Gilks, W. and Berzuini, C. (2001), Following a moving target -Monte Carlo inference for dynamic Bayesian models, *J. Roy. Statist. Soc. B* 63:127–146.
8. Chorin, A.J. and Krause, P. (2004), Dimensional reduction for a Bayesian filter. *Proc. Nat. Acad. Sci. USA* 101:15013–15017.
9. Dowd, M. (2006), A sequential Monte Carlo approach for marine ecological prediction, *Environmetrics* 17:435–455.
10. Doucet, A. and Johansen, A., Particle Filtering and Smoothing: Fifteen years Later, Handbook of Nonlinear Filtering (eds. D. Crisan et B. Rozovsky), Oxford University Press, to appear.
11. Snyder, C., Bengtsson, T., Bickel, P, and Anderson, J. (2008), Obstacles to high-dimensional particle filtering, *Mon. Wea. Rev.* 136:4629–4640.
12. Bickel, P, Li, B., and Bengtsson T. (2008), Sharp failure rates for the bootstrap particle filter in high dimensions, *IMS Collections: Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh* 3:318-329.
13. Chorin, A.J. (2008), Monte Carlo without chains, *Comm. Appl. Math. Comp. Sc.* 3:77–93.
14. Weare, J. (2007), Efficient Monte Carlo sampling by parallel marginalization, *Proc. Nat. Acad. Sc. USA* 104:12657–12662.
15. Weare, J. (2009), Particle filtering with path sampling and an application to a bimodal ocean current model, *J. Comput. Phys.* 228: 4321-4331.
16. Gordon, N., Salmond, D., and Smith A. (1993), Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEEE Proceedings-F* 140: 107–113.
17. Carpenter, J., Clifford, P., and Fearnhead, P. (1999), An improved particle fiter for nonlinear problems, *IEEE Proceedings-Radar Sonar and Navigation* 146:2–7.
18. Milstein, G., Platen, E., and Schurz H. (1998), Balanced implicit methods for stiff stochastic systems, *SIAM J. Num. Anal.* 35:1010–1019.
19. Kloeden, P. and Platen, E. (1992), Numerical Solution of Stochastic Differential Equations, Springer, Berlin.
20. Stuart, A., Voss, J., and Wilberg, P. (2004), Conditional path sampling of SDEs and the Langevin MCMC method, *Comm. Math. Sc.* 4:685–697.
21. Levy, P. (1954), Le Mouvement Brownien, Gauthiers-Villars, Paris.