Series expansions of Brownian motion and the unscented particle filter

October 15, 2013

Abstract

The discrete-time filtering problem for nonlinear diffusion processes is computationally intractable in general. However, it is relatively easy to draw approximate samples from a nonlinear diffusion process. For this reason, methods such as the bootstrap filter are particularly effective at approximating the optimal filtering distribution.

In this paper, we first make an approximation to the signal process \( X \) by discarding the ‘high frequency’ components of the Brownian motion \( W \) driving the signal. We show that one can modify the remaining random coefficients using importance sampling, so that the particles in the filter are guided toward regions of high likelihood.

We first use the unscented transform to find a Gaussian approximation to the joint distribution of the Fourier coefficients and the observations. The Gaussian approximation is then conditioned on incoming observations, yielding an importance distribution for the Fourier coefficients. We show that performance of the particle filter improves when we use this importance distribution.

1 Introduction

Sequential Monte-Carlo (SMC) methods are a powerful and general family of techniques for generating samples from high-dimensional probability distributions. Among the best-known applications of SMC is the particle filter. In this paper, we discuss a novel way of applying a particle filter when the signal of interest is a nonlinear multivariate diffusion process.

The continuous-discrete filtering problem consists of estimating some property of a signal process \( X \) given a set of noisy discrete-time observations \( \{Y_i\} \). It may be the case that we only observe certain components of the signal, or that we observe the image of the signal under a nonlinear transform.

The filtering problem for nonlinear diffusion processes was studied by Kushner [1] and Zakai [2], among many others. The general solution of the problem is given in terms of a stochastic partial differential equation, which is computationally intractable in the general case. Particle filters offer a promising means
of approximating the optimal solution of the filtering problem. A number of excellent tutorials and reviews are available: see, for example, [3] [4] [5], and [6] for a textbook-length collection of important developments up to 2001.

Roughly speaking, the particle filter works as follows. At time $t_{k-1}$, we assume we have a collection of weighted point masses, or ‘particles’ that approximate the filtering distribution. The particles are propagated forward in time using a suitable transition density, resulting in a new collection of point masses. These are assigned a new set of weights by means of importance sampling. The weights are computed using knowledge of the signal dynamics, the observation at time $t_k$ and the importance transition density.

The bootstrap filter [7] is one simple implementation of a particle filter. In the bootstrap filter, one uses the prior dynamics of the signal to propagate the particles forward in time. The bootstrap filter can fail when the conditional distribution $p(X_{t_k} | Y_{t_k}, X_{t_{k-1}})$ is tightly constrained relative to the unconditional distribution $p(X_{t_k} | X_{t_{k-1}})$. This is because most of the particles are assigned a low weight in the importance sampling step. In the worst case, one approximates the filtering distribution with what amounts to a single particle. This phenomenon tends to occur in high-dimensional filtering problems, or in situations where the observations are highly informative about the state of the system [8].

One way of correcting for the degeneracy issue is to employ better importance distributions. To form the optimal importance distribution, one would ideally compute the joint distribution of the signal and the observations at time $t_k$. One could then use the joint distribution to compute the distribution of the signal conditional on the observation. In practice, due to nonlinearities in the system, this cannot be done efficiently. However, one can construct a Gaussian approximation to the joint distribution of the signal and the observations at time $t_k$. One can then use standard results about the multivariate normal distribution to approximate the conditional distribution. The conditioned distribution is often an effective importance distribution for the particle filter.

The Gaussian approximation can be constructed by Taylor expanding the signal and observation functions to first order. Over short timescales, or for processes whose dynamics are ‘almost’ linear, this can be an effective strategy. However, the ‘almost linear’ assumption does not always hold, and performance of the Taylor method can be poor. An alternative is to use the unscented transform [9] [10] [11], which often produces a more accurate approximation. This is known as the Unscented particle filter [12].

When the underlying signal is a diffusion process, it is often impossible to compute the density $p(X_{t_k} | X_{t_{k-1}})$. For this reason, the importance distributions that one can employ are rather restricted, since in most cases one must rely on some sort of cancellation from the importance distribution. In this paper, the diffusion process at time $t_k$ is approximated as the image of a Gaussian distribution under a certain nonlinear transform $f$. That is, $X_{t_k} \approx f(X_{t_{k-1}}, Z)$. We use the unscented transform to construct an importance variate $V$ such that $f(X_{t_{k-1}}, V)$ sidesteps the particle degeneracy issue while still producing tractable importance weights.
A number of other methods have been applied to the problem. Fearnhead et al. [13] [14] use a form of rejection sampling to construct an unbiased estimator of the importance weights. This obviates the need for some term in the importance distribution to cancel with \( p(X_{t_k}|X_{t_{k-1}}) \), allowing the authors to employ more general importance distributions. However, the algorithm is only applicable to certain classes of diffusion process, and it only performs well when the variance of the estimator is sufficiently low.

Rimmer et al. [15] discuss the use of a particle filter for maximum likelihood estimation of some parameter governing the dynamics of the diffusion process. Use of the unscented transform within a discrete approximation to the diffusion is discussed. Our use of the unscented transform is different, and considerably cheaper in terms of computational cost.

Murray and Storkey [16] use a high-order numerical integration scheme to approximate both filtering and smoothing sensitivities by means of a forward-backward recursion. Practical applications of the particle filter to fMRI data are discussed in [17].

Other approaches typically use the Girsanov theorem to construct importance processes that have a tractable density with respect to the law of the signal \( X \). Maroulas and Stinis [18] use ‘drift homotopy’ (a method related to simulated annealing) to find ’good’ importance processes. Särkkä and Sottinen suggest linearising the SDE to construct an importance process with the appropriate mean and covariance.

The rest of the paper is structured as follows. In Section 2, we describe the filtering problem in more detail. We describe some existing particle filters such as the bootstrap filter and unscented particle filter. In Section 3, we review the series expansion approximation of Brownian motion, and show how it can be used in the context of importance sampling. In Section 3.1 we describe the series expansion unscented particle filter. Section 5 provides some numerical experiments, and we discuss our findings and some possibilities for future work in Section 6.

2 Particle filtering of diffusion processes

In this section, we will briefly review the ideas behind sequential importance sampling, and particle filtering in particular. We will assume that our signal, \( X \), is a diffusion process satisfying the stochastic differential equation

\[
dX_t = a(X_t)dt + b(X_t)dW_t, \quad X_0 \sim p_{X_0}. \tag{1}
\]

Here, \( X_t \in \mathbb{R}^n \), \( W_t \in \mathbb{R}^d \), \( a: \mathbb{R}^n \to \mathbb{R}^n \), and \( b: \mathbb{R}^n \to \mathbb{R}^{n \times d} \). We assume \( a \) and \( b \) satisfy the usual conditions that ensure \( X \) has a unique strong solution (for example, the assumption that they are globally lipschitz is strong enough).

We assume that noisy observations of the process \( X \) are made at times \( t_{1:k} \). We will typically use the notation \( X_{t_{1:k}} \) to refer to the sequence of random variables \( X_{t_1}, X_{t_2}, \ldots, X_{t_k} \). This enables us to use notation that is consistent with what is usually seen in the literature on discrete-time particle filtering.
Observations are assumed to arrive in the form
\[ Y_{tk} = h(X_{tk}) + \epsilon_{tk}, \]  
where \( h : \mathbb{R}^n \to \mathbb{R}^m \), and \( Y_t \in \mathbb{R}^m \). The random variables \( \{\epsilon_t\} \) are jointly Gaussian and independent, each with mean 0 and covariance \( R \). We assume a fixed time \( T = t_k - t_{k-1} \) between observations for clarity of exposition, though this is easy to generalise.

Diffusion processes possess the Markov property, so that the joint distribution of state and observations factorises as
\[ p(X_{1:k}, Y_{1:k} | X_0) = \prod_{i=1}^{k} p(Y_{ti} | X_{ti})p(X_{ti} | X_{ti-1}). \]  
In order to compute the transition density \( p(X_{tk} | X_{tk-1}) \) for a general nonlinear diffusion, one must solve a partial differential equation known as the Fokker-Planck equation. It is only in special cases that this solution is available in closed form, and one must typically resort to approximations of some sort.

In the Bayesian filtering paradigm, our aim is to compute the posterior distribution of the state of \( X_{tk} \) given the observations \( Y_{1:k} \) and some prior distribution on the initial state \( X_0 \). This can be done iteratively as follows. We divide the filtering problem up into two phases: the prediction phase, and the update phase. In the predictive step, we compute or approximate the distribution
\[ p(X_{tk} | Y_{1:k-1}) = \int p(X_{tk} | X_{tk-1})p(X_{tk-1} | Y_{1:t-1})dX_{tk-1}. \]  
At time \( t_k \), we make the observation \( Y_{tk} \). This can be incorporated into the posterior distribution via an application of Bayes’ theorem as follows:
\[ p(X_{tk} | Y_{1:k-1}, Y_{tk}) = \frac{p(Y_{tk} | X_{tk}, Y_{1:k-1})}{p(Y_{tk} | Y_{1:k-1})} = \frac{p(Y_{tk} | X_{tk})p(X_{tk} | Y_{1:k-1})}{p(Y_{tk} | Y_{1:k-1})}. \]  
The prediction step relies on the intractable transition density \( p(X_{tk} | X_{tk-1}) \). In addition, the normalising constant \( p(Y_{tk} | Y_{1:k-1}) \) in the update step is typically difficult to compute when the state dimension of \( X \) is large. In the next section, we will see how importance sampling can be exploited to overcome both of these issues.

2.1 Sequential importance sampling and the bootstrap filter
As we noted earlier, it is usually impractical to evaluate the transition density of a general diffusion process at a given time. However, it is relatively straightforward to construct an approximate sample path of a diffusion process. This
can be achieved by discretising the process and applying an numerical scheme such as, for example, the Euler-Maruyama method or the Ito-Taylor scheme. In some cases it is even possible to draw sample paths from a nonlinear diffusion without inducing any discretisation bias [13] [14].

We will now show how sequential importance sampling (SIS) can be applied to a diffusion process in order to compute an approximate solution to the filtering problem. Monte-Carlo methods such as SIS are particularly suited to filtering problems involving diffusion processes due to the relative ease of drawing sample paths, and the presence of the Markov property.

Suppose at time $t_{k-1}$, we have a collection of ‘particles’ $\{X_{i_{t_{k-1}}}\}$ and positive weights $\{w_{i_{t_{k-1}}}\}$ that sum to unity, such that the filtering distribution $p(X_{t_{k-1}}|Y_{t_{1:k-1}})$ is approximated by the weighted Monte-Carlo estimate

$$p(X_{t_{k-1}}|Y_{t_{1:k-1}}) \approx \sum_{i} w_{i_{t_{k-1}}} \delta_{X_{i_{t_{k-1}}}^{t_{k-1}}}.$$  \hspace{1cm} (6)

A weighted sample such as this can be initialised at time 0 by drawing uniformly-weighted samples from the prior distribution on $X_0$.

We now apply the prediction and update equations (4) and (5) to the particle representation (6). This will give us a representation of the time-$t_k$ filtering distribution in terms of the observation and transition densities, and the particles from (6).

Applying the prediction equation, we find that

$$p(X_{t_k}|Y_{t_{1:k-1}}) \approx \sum_{i} p(X_{t_k}|X_{i_{t_{k-1}}}) w_{i_{t_{k-1}}}.$$  \hspace{1cm} (7)

The predictive distribution is updated with the observation at time $t_k$, yielding

$$p(X_{t_k}|Y_{t_{1:k}}) \approx \frac{1}{p(Y_{t_k}|Y_{t_{1:k-1}})} \sum_{i} p(Y_{t_k}|X_{i_{t_k}}) p(X_{t_k}|X_{i_{t_{k-1}}}) w_{i_{t_{k-1}}}.$$  \hspace{1cm} (8)

This distribution is also intractable due to the presence of the transition density $p(X_{t_k}|X_{i_{t_{k-1}}})$. However, we can generate a new weighted particle approximation of $p(X_{t_k}|Y_{t_{1:k}})$ by an application of importance sampling.

We begin by sampling particles $\{X_{i_{t_k}}\}$ from an importance distribution $q(X_{i_{t_k}}|X_{i_{t_{k-1}}}, Y_{t_k})$. One can avoid computing the normalising constant in (7) by first computing unnormalised weights $\{\tilde{w}_{i_{t_k}}\}$ for the particles $\{X_{i_{t_k}}\}$. The unnormalised weights are given by

$$\tilde{w}_{i_{t_k}} = w_{i_{t_{k-1}}} p(Y_{t_k}|X_{i_{t_k}}) p(X_{i_{t_k}}|X_{i_{t_{k-1}}}) q(X_{i_{t_k}}|X_{i_{t_{k-1}}}, Y_{t_k}).$$  \hspace{1cm} (9)

The weights are normalised by setting $w_{i_{t_k}} = \tilde{w}_{i_{t_k}} / \sum \tilde{w}_{i_{t_k}}$. Finally, we arrive at the new weighted particle approximation

$$p(X_{t_k}|Y_{t_{1:k}}) \approx \sum_{i} w_{i_{t_k}} \delta_{X_{i_{t_k}}}.$$  \hspace{1cm} (10)
Repeated application of the weight update equation (9) as described above can be problematic. Typically, one of the weights will grow to dominate the others, with the result that the filtering distribution is effectively approximated by a single particle. This issue can be resolved by adding a resampling step. When the weights become sufficiently imbalanced, we create a new, uniformly weighted collection of particles by sampling from the discrete distribution in (6) with replacement. The prediction and update steps are then applied as usual.

We still need to deal with the appearance of the transition density in the weight update equation (9). The importance distribution

\[ q(X_{t_k}|X_{t_{k-1}}, Y_{t_k}) \]

can be chosen to cancel with this term. One simple way of achieving this is to set

\[ q(X_{t_k}|X_{t_{k-1}}, Y_{t_k}) = p(X_{t_k}|X_{t_{k-1}}). \]

This is known as the bootstrap filter.

2.2 The unscented particle filter

Doucet, et al. [19] have shown that the optimal importance distribution in (9) is

\[ q(X^i_{t_k}|X^i_{t_{k-1}}, Y_{t_k}) = p(X^i_{t_k}|X^i_{t_{k-1}}, Y_{t_{k}}). \]  \( (11) \)

That is, the optimal importance distribution at time \( t_k \) is the filtering distribution conditioned on the incoming observation at time \( t_k \). The distribution is optimal in the sense that it minimises variance of the importance weights. It also has the property that the importance weights at time \( t_k \) depend only on the state of the particle at time \( t_{k-1} \).

It is often infeasible to sample from this distribution, though one can attempt to sample from a distribution that approximates it. One method of doing so uses the unscented transform to approximate the mean and variance of the optimal importance distribution. A Gaussian distribution with the corresponding mean and variance is used in place of the optimal distribution. This is known as the unscented particle filter [20].

We now present a short review of the concepts underlying the unscented transform.

2.2.1 The unscented transform

Suppose we are given a random variable \( X \) and a nonlinear function \( f \). The distribution of \( f(X) \) can be computed in terms of the inverse of \( f \) and its Jacobian. However, it can be cumbersome to compute or approximate this distribution. One is often interested in the first few moments of \( f(X) \) rather than the full distribution. The unscented transform provides a computationally efficient means of computing moments of \( f(X) \). A number of ‘sigma points’ are chosen in order to attempt to capture the statistics of \( X \). The sigma points are propagated through the function \( f \), and the moments of \( f(X) \) are approximated by a weighted sum of the image points. The unscented transform is similar to a Monte-Carlo approximation of \( f(X) \). The key difference is that the sigma points are chosen in a principled manner, with the aim of minimising the number of evaluations of \( f \) that must be made.
We will restrict our exposition to the case where $X$ has an $n$-dimensional multivariate normal distribution, and we wish to fit a multivariate normal distribution to $f(X)$. Suppose $X$ has mean $m$ and covariance $P$. The unscented transform uses $2n + 1$ sigma points, which are constructed as follows. One chooses two tuning parameters $\alpha$ and $\kappa$, then sets $\lambda = \alpha^2(n + \kappa) - n$. The sigma points are then defined by the following expressions:

$$\sigma^0 = m,$$

$$\sigma^i = m + (\sqrt{(n + \lambda)}\mathbf{P})_{si}, \quad 1 \leq i \leq n,$$

$$\sigma^{n+i} = m - (\sqrt{(n + \lambda)}\mathbf{P})_{si}, \quad 1 \leq i \leq n.$$  

The sigma points are determined once one chooses an appropriate matrix square root. Here $(\sqrt{\mathbf{P}})_{si}$ is the $i$-th column of the matrix square root of $\mathbf{P}$ defined via $\mathbf{P} = \sqrt{\mathbf{P}}\sqrt{\mathbf{P}}^\top$.

The mean and covariance of $f(X)$ are approximated by a weighted average of the sigma-point images. Define $Y_i = f(\sigma^i)$, and set

$$\mathbb{E}[f(X)] \approx \mu = \sum_{i=0}^{2n} w^{(m)}_i Y_i.$$  

We can then make the approximations

$$\text{Cov}[f(X)] \approx \mathbf{S} = \sum_{i=0}^{2n} w^{(c)}_i (Y_i - \mu) (Y_i - \mu)^\top,$$

$$\text{Cov}[X, f(X)] \approx \mathbf{C} = \sum_{i=0}^{2n} w^{(c)}_i (\sigma^i - m) (Y_i - \mu)^\top.$$  

The weights depend on a third tuning parameter $\beta$, which controls the accuracy of higher-order moments in the transform [21] (in the Gaussian case, $\beta = 2$ is optimal). The weights are given by

$$w^{(m)}_0 = \frac{\lambda}{n + \lambda},$$

$$w^{(c)}_0 = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta),$$

$$w^{(m)}_i = \frac{1}{2(n + \lambda)} \quad i = 1, \ldots, 2n,$$

$$w^{(c)}_i = \frac{1}{2(n + \lambda)} \quad i = 1, \ldots, 2n.$$  

It is well known that the unscented transform matches the mean of $f(X)$ exactly when $f$ is a polynomial of degree three or less. In general, errors in the estimate of the mean are introduced only by the fourth and higher terms in the Taylor expansion of $f$ [11].
3 Brownian Series expansions and stochastic differential equations

We now describe a method for obtaining a smooth approximation of Brownian motion by decomposing it in a generalised Fourier series. We aim to use the smooth approximation as a driving function in a differential equation. This will enable us to approximate a nonlinear stochastic differential equation with a randomised ordinary differential equation, which will prove to be computationally tractable to work with. This approximation was used as the basis of a Markov chain Monte Carlo algorithm for Bayesian parameter estimation of a nonlinear diffusion in [22].

Suppose $W = (W^{(1)}, \ldots, W^{(d)})$ is a standard $d$-dimensional Brownian motion, and let $\{\phi_i\}_{i \geq 1}$ be an orthonormal basis of $L^2([0,T], \mathbb{R})$. We use the notation $I$ to denote the indicator function. That is, $I_{[0,t]}(u) = 1$ when $0 \leq u \leq t$, and $I_{[0,t]}(u) = 0$ otherwise. One can construct a series expansion of $W$ in terms of the basis functions $\{\phi_i\}$ as follows [23]:

$$W_t = \int_0^T I_{[0,t]}(u)dW_u$$
$$= \int_0^T \sum_{i=1}^{\infty} \langle I_{[0,t]}, \phi_i \rangle \phi_i(u) dW_u$$
$$= \sum_{i=1}^{\infty} \left( \int_0^T \phi_i(u)dW_u \right) \left( \int_0^t \phi_i(u)du \right). \tag{19}$$

We use the standard inner product on $L^2[0,T]$, which is defined as

$$\langle f, g \rangle = \int_0^T f(u)g(u)du. \tag{20}$$

The stochastic integrals are i.i.d $d$-dimensional standard normal. We can see this by noting that he basis functions are deterministic,

$$\mathbb{E} \left[ \int_0^T \phi_i(u)dW_u \right] = 0, \tag{21}$$

and

$$\text{Cov}(Z_i, Z_j)$$
$$= \mathbb{E} \left[ \left( \int_0^T \phi_i(u)dW_u \right) \left( \int_0^T \phi_j(u)dW_u \right)^\top \right]$$
$$= \left( \int_0^T \phi_i(u)\phi_j(u)du \right) I_d = \delta_{ij}I_d. \tag{22}$$
Here, $I_d$ is the $d \times d$ identity matrix.

For ease of notation, we set

$$Z_i = \int_0^T \phi_i(u)dW_u.$$  \hspace{1cm} (23)

We conclude that

$$W_t = \sum_{i=1}^{\infty} Z_i \int_0^t \phi_i(u)du.$$  \hspace{1cm} (24)

We can obtain an approximation of a Brownian sample path by drawing i.i.d samples $Z_i$ from a standard normal distribution and truncating the sum in (24). This allows us to describe a Brownian sample path approximately in terms of a finite number of variates.

We will adopt the convention that the basis functions are ordered by the number of times they change sign on the interval $[0,T]$. Thus, when $i$ is small, $Z_i$ governs low-frequency oscillations of the Brownian motion. If we observe $W$ at discretely-spaced timesteps, we should not expect to be able to infer much information about the very high frequency terms: this follows from Nyquist’s theorem and related results. For our purposes, the most ‘interesting’ coefficients will be those that govern the low-frequency terms.

### 3.1 Series Expansion Approximation of SDE

In order to approximate the diffusion $X$, we truncate the series expansion (24) after $N$ terms, and use the resulting smooth process as an approximation of Brownian motion. In other words, we make the approximation

$$dW_t \approx \sum_{i=1}^{N} Z_i \phi_i(t)dt.$$  \hspace{1cm} (25)

We will refer to the process driven by the right-hand side of (25) as $\hat{X}$. Since $\hat{X}$ is driven by a finite linear combination of basis functions, the resulting process is differentiable. We can therefore interpret $\hat{X}$ as the solution to an ordinary differential equation, which satisfies

$$\frac{d\hat{X}_t}{dt} = a(\hat{X}_t) + \sum_{i=1}^{N} b(\hat{X}_t)Z_i \phi_i(t), \quad \hat{X}_0 = x_0.$$  \hspace{1cm} (26)

Approximations of this type were first investigated by Wong and Zakai [24], who showed that in the one-dimensional case, $\hat{X}_t$ converges to the Stratonovich solution of the stochastic differential equation [25].

In the multivariate case with state-dependent diffusion coefficient, one may need to place additional constraints on the sequence of Brownian approximations in order to ensure convergence to the correct limit. Sufficient conditions are given in [26]. The Haar wavelets are one set of basis functions known to
possess the necessary properties for convergence to the Stratonovich limit. If the diffusion coefficient does not depend on the state of the system, then the ‘stochastic correction’ term disappears and convergence is guaranteed.

The approximation (26) has the advantage of re-casting an infinite dimensional problem in finite-dimensional terms. Given the value of $X_{t_{k-1}}$ as an initial condition, we can view the solution of (26) as a function

$$\hat{X}_{t_k} = f(T, X_{t_{k-1}}, Z_{1:N}).$$

Here, $f$ solves the ordinary differential equation (26), and $\{Z_i\}_{1 \leq i \leq N}$ are i.i.d standard $d$-dimensional Gaussian random variables. In essence, the time-$t$ distribution of the process $\hat{X}$ can be interpreted as the image of a Gaussian distribution under a nonlinear transform. This is precisely the setting for which sigma-point methods were designed.

Of course, one can apply this reasoning to other methods for approximating SDEs. In the $N$-step Euler-Maruyama scheme, one sets $\Delta t = T/N$, and

$$X_{(n+1)\Delta t} = g(X_{n\Delta t}, Z_n) = X_{n\Delta t} + a(X_{n\Delta t})\Delta t + b(X_{n\Delta t})Z_n\sqrt{\Delta t}.$$  \hspace{1cm} (28)

The equivalent of $f$ in (27) is the $N$-fold application of $g$ to the initial state $X_0$. However, this approximation puts all random variables $\{Z_i\}$ on an ‘equal footing’, and loses the interpretation of the coefficients as controllers of the behaviour of the SDE at varying frequencies.

### 3.2 SDE approximations and importance sampling

In the previous section, we saw that the value of $X_{t_k}$ can be approximated as the image of a collection of i.i.d. Gaussian random variables (along with a possibly random initial condition) under a nonlinear transform. For the purposes of describing the SE-UPF, we assume the initial condition is fixed, though this can easily be generalised.

One is often interested in computing expectations of the form

$$\mathbb{E}[g(X_{t_k})] \approx \mathbb{E}[g(\hat{X}_{t_k})] = \mathbb{E}\left[g\left(f(T, X_{t_{k-1}}, Z_{1:N})\right)\right],$$

where $f$ is the solution of the ODE (26) described in Section 3.1.

In cases where the expectation cannot be computed exactly, one natural way to proceed is to draw a number of i.i.d samples $\{\hat{X}_{t_k}^i\}_{i \leq n}$ from $\hat{X}_{t_k}$ and approximate the expectation via the Monte-Carlo estimate

$$\mathbb{E}[g(\hat{X}_{t_k})] \approx \frac{1}{n} \sum_{i=1}^{n} g(\hat{X}_{t_k}^i)$$

$$= \frac{1}{n} \sum_{i=1}^{n} g\left(f(T, X_{t_{k-1}}, Z_{1:N})\right)$$

The right-hand side of (30) is an unbiased estimator of $\mathbb{E}[g(\hat{X}_{t_k})]$. However, its variance may be unacceptably high. Consider, for example, a function $g$ that takes a large value somewhere in the tails of the distribution of $X_{t_k}$. 
To work around this issue, one can attempt to construct a more suitable collection of 'importance' random variables $Z_{1:N} \sim q(\cdot)$. We can perform importance sampling by drawing i.i.d samples $\{Z_{1:N}^i\}$ from $q(\cdot)$ and weighting the Monte-Carlo estimate appropriately:

$$
E[g(\hat{X}_{tk})] \approx \frac{1}{n} \sum_{i=1}^{n} g \left( f(t, X_{tk-1}, Z_{1:N}^i) \right) \frac{N(Z_{1:N}^i|0, I_{Nd \times Nd})}{q(Z_{1:N}^i)}. 
$$

(31)

If the distribution $q$ is chosen appropriately, one can achieve a reduction in the variance of the estimator. We will use this methodology to 'guide' particles in the SE-UPF toward regions of high likelihood.

4 The series expansion unscented particle filter

We will now describe one way of applying the ideas described in Section 3.2 to the particle filtering framework. For a general nonlinear diffusion, it is not feasible to compute the transition density $p(X_{tk}|X_{tk-1})$. Instead of working directly with the density of $X_{tk}$, we opt instead to use importance sampling to modify the driving noise. In terms of the SDE approximation derived in Section 3.1, the expression for the unnormalised weight update (9) is

$$
\tilde{w}_{tk}^i = w_{tk-1}^i \frac{p(Y_{tk}|f(t, X_{tk-1}^i, Z_{1:N}^i), N(Z_{1:N}^i|0, I_{Nd \times Nd})/q(Z_{1:N}^i), Y_{tk})}{q(Z_{1:N}^i|X_{tk-1}^i, Y_{tk})},
$$

(32)

Here, $q$ is playing the role of the importance distribution in (9), though we have modified the driving noise of $X$ instead of modifying the distribution of $X_{tk}$ directly. As we will see in the next section, the unscented transform provides a computationally inexpensive means of constructing useful importance distributions for the driving noise.

4.1 Choice of importance distribution

There are several ways in which one could form an appropriate importance distribution. One possible way to proceed is to construct a Gaussian approximation of the joint distribution $p(Y_{tk}, Z_{1:N}|X_{tk-1})$. Given such an approximation, standard results about the multivariate normal distribution allow us to compute a Gaussian approximation to $p(Z_{1:N}|Y_{tk}, X_{tk-1})$, which we will use as our importance distribution $q(\cdot)$.

In what follows, we will use the unscented transform to construct a Gaussian approximation of the joint distribution of $Y_{tk}$ and $Z_{1:N}$. The random variable $Z_{1:N}$ is $N \times d$-dimensional standard normal. We select $2Nd + 1$ sigma points $\{\sigma^j\}$ to capture the mean and covariance of $Z_{1:N}$.

For a suitable initial condition $X^{ic}$ (e.g. the ensemble mean, or the location of a given particle $X_{tk-1}^i$), we set

$$
Y^j = h(f(X^{ic}, \sigma^j)), \quad 1 \leq j \leq 2Nd + 1.
$$

(33)
We can apply equations (15), (16) and (17) with the appropriate weighting to find \( \mu, S \) and \( C \) (recall that these quantities approximate the mean of \( Y \), the covariance of \( Y \) neglecting observation noise, and the cross-covariance of \( Z_{1:N} \) and \( Y \) respectively). When we account for the additional variance \( R \) added by the observation noise, we obtain

\[
\text{Cov}(Y_{t_k}, Z_{t_k}) \approx \left( S + R \right) C^T I_N.
\]

The approximate mean and variance of \( Z_{1:N} \) conditional on \( Y_{t_k} \) is then

\[
E[Z_{1:N}|Y_{t_k}, X_{t_k}] \approx \mu_Y = C(S + R)^{-1}(Y_{t_k} - \mu),
\]

\[
\text{Cov}(Z_{1:N}|Y_{t_k}, X_{t_k}) \approx \Sigma_Y = I_N - C(S + R)^{-1}C^T.
\]

We can use these quantities as a basis for constructing an importance distribution \( q \). The simplest method is simply to set \( q = \mathcal{N}(\mu_Y, \Sigma_Y) \). However, theoretical analysis of importance sampling shows that one should employ a distribution with heavier tails than the target distribution. One possibility is to set the diagonal entries of \( \Sigma_Y \) to 1.

In order to compute \( \mu_Y \) and \( \Sigma_Y \), we must solve \( 2Nd + 1 \) differential equations: one for each sigma point. This can be done for the ensemble of particles as a whole, or for each unique particle individually. That is, \( \mu_Y \) and \( \Sigma_Y \) may be computed ‘globally’ using some summary statistic of the ensemble, or ‘locally’, depending on each individual particle. For the ‘global’ approach, we use the ensemble mean as an initial condition in (33). This represents an increase in cost over the bootstrap filter that is independent of the number of particles. For the local approach, we use the location of each of the \( n \) particles as the initial condition in (33).

The ‘local’ method requires that (33) be solved for each individual particle. Naively, one might think that this entails solving the ODE with \( n \) distinct initial conditions. However, after resampling, many of the particles share the same location. In practice, we need to solve significantly fewer than \( n \) systems of ODEs. The increase in performance cost of the ‘local’ method is smallest in situations that standard particle filters struggle to cope with due to low effective sample size.

In practice, we find that the posterior mean and covariance tend only to change for the first few coefficients \( Z_{1:M} \), with the distribution of the remainder being similar to the prior (i.e. standard normal). Intuitively speaking, this is sensible. The ‘high frequency’ basis functions control the behaviour of the SDE at small length-scales. In the typical case, one should not expect to be able to deduce small-scale behaviour from observations spaced far apart in time.

Since observations tend not to be informative about higher-frequency components, we can perform importance sampling guided by the unscented transform on a small number of coefficients \( Z_{1:M} \), and draw the remaining coefficients \( Z_{M+1:N} \) from the prior. The value of \( M \) depends on the nature of the model,
but in the high-dimensional experiment below, we saw good results with $M = 2$, $N = 4$.

## 5 Numerical experiments

### 5.1 Sinusoidal diffusion

For our first test, we reproduced the setup from one of the experiments in [13]. A sample path was drawn from the one-dimensional diffusion

$$dX_t = \sin(X_t)dt + dW_t, X_0 = 0.$$  \hfill (37)

We made 50 noisy observations of the process, spaced $T = 1$ units of time apart. The variance of the observation noise was set to $R = .5$. The performance of the SE-UPF was compared against that of the bootstrap filter (using an Euler-Maruyama discretisation) and the random-weight particle filter from [13]. The importance distribution for the RWPF used the ensemble mean (i.e. the same distribution was used for each particle). We assess the performance of the filters by measuring the variance their estimate of the filter mean over several independent runs, having fixed a sample path and a set of observations. We assume that filters with a lower variance are performing better.

It is not straightforward to devise a simple and fair way of comparing these filters. The random weight filter makes an unbiased estimate of the importance sampling weights, whereas both the SE-UPF and Euler-Maruyama bootstrap filter are subject to approximation bias. Computational cost per particle is not necessarily an ideal metric. One can choose a large timestep in an Euler scheme or when solving (26) to propagate particles forward in time cheaply. Provided that one is prepared to tolerate this bias, these methods will dominate the RWPF. In addition, one or other algorithm could be heavily optimised to boost its performance.

Another possibility is to use a set number of particles and compare the performance across algorithms without regard to computational cost. Again, this is not ideal since any useful improvement to a particle filter should be expected to outperform the baseline method of simply adding more particles to the bootstrap filter up to a computational cost equivalent to that of the improvement.

For sufficiently low observation noise (roughly $R < .3$), the RWPF outperformed the SE-UPF and Euler-Maruyama bootstrap filters for a fixed number of particles ($n = 500$). However, the RWPF was more computationally expensive. For $R > .3$, the SE-UPF outperformed the RWPF and bootstrap filters on a per-particle basis. It was cheaper than the RWPF, and the bootstrap filter. It is surprising that the SE-UPF would be cheaper than the bootstrap filter. This is because the standard Euler-Maruyama scheme needs to generate draws from a standard normal distribution for each timestep, whereas the SE-UPF needs fewer such draws. The relative variance of the estimates of the filter mean are shown in Figure 1.
5.2 Coordinated turn model

For our next experiment, we simulated a number of sample paths from a model of an aircraft performing a 'coordinated turn'. The state of the aircraft is modelled using a seven dimensional SDE. We use use subscripts to denote individual components of the state vector. For example, \( X_{1:3}(t) \) refers to the first, second and third components at time \( t \). The state vector of the aircraft records position, velocity and the rate of turning. The dynamics are given by

\[
dX(t) = a(X(t))dt + b(X(t))dW_t, \tag{38}
\]

with

\[
a(x_{1:7}) = \begin{pmatrix} x_2 \\ -x_7x_4 \\ x_4 \\ x_7x_2 \\ x_6 \\ 0 \\ 0 \end{pmatrix} \tag{39}
\]
and

\[ b(x_{1:7}) = \begin{pmatrix}
0 & 0 & 0 & 0 \\
\sqrt{1+x_1^2} & \sqrt{1+x_2^2} & \sqrt{(1+x_1^2)(1+x_2^2)} & 0 \\
0 & 0 & 0 & 0 \\
\sqrt{1+x_3^2} & \sqrt{1+x_4^2} & \sqrt{(1+x_3^2)(1+x_4^2)} & 0 \\
0 & 0 & 0 & 0 \\
\sqrt{1+x_5^2} & 0 & -v_{xy} & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}. \tag{40}\]

Here, \( v = \sqrt{1+x_3^2+x_4^2+x_6^2} \) and \( v_{xy} = \sqrt{1+x_2^2+x_4^2} \). While the state is seven-dimensional, we only need a four-dimensional driving Brownian motion since the diffusion matrix \( b \) is not of full rank. We assume the covariance of this driving Brownian motion is a diagonal matrix \( Q \).

Components 1, 3 and 5 of the state vector represent the position of the aircraft, and are measured in meters. Components 2, 4 and 6 represent the velocity and are measured in m/s. The seventh component measures the turn rate of the vehicle and is measured in 100ths of a degree per second.

We set \( X_0 = (1000, 100, 1000, 100, 1000, 100, 600) \), \( Q = \text{diag}(100, 100, 100, 1000) \), and the inter-observation time was set to \( T = 1 \). We modelled observations as arriving through a ground-based radar dish that can determine range, azimuthal angle, and attitudinal angle. For this reason, only the positional co-ordinates \((x_1, x_3, x_5)\) were observable. The observation function is given by

\[ h(x_{1:7}) = \begin{pmatrix}
\sqrt{x_1^2 + x_3^2 + x_6^2} \\
\tan^{-1}(x_3/x_1) \\
\tan^{-1}(x_5/\sqrt{x_1^2 + x_2^2})
\end{pmatrix}. \tag{41}\]

The observation error was assumed to have covariance \( R = \text{diag}([1, 0.005, 0.005]) \).

We tested the SE-UPF against the bootstrap filter. The RWPF is not applicable in this situation since the diffusion matrix is not of full rank. To be as fair as possible, we used the same code for the bootstrap filter as for the SE-UPF. The bootstrap filter code was modified to draw \( Z_{1:N} \) from the prior distribution instead of using an importance distribution.

Both tests used approximately the same amount of processor time: for the standard particle filter, we used \( n = 450 \) particles. For the series expansion filter, we used \( n = 200 \) particles. We drew a sample path from \( X \) and ran both filters 500 times. For each filter, we computed \( \text{Var}(\hat{m}_t) \), the variance in the estimate of the filtering mean at time \( t \). A large variance between runs of a filter indicates that it is a poor approximation of the optimal filter.

We computed \( \text{Var}(\hat{m}^{SE}_t) \) and \( \text{Var}(\hat{m}^B_t) \), respectively the sample variance in the series expansion filter and bootstrap filter estimate of \( E[X_t|Y_{1:t}] \). In Figure 3, we plot \( \text{Var}(\hat{m}^{SE}_t)/\text{Var}(\hat{m}^B_t) \) as a function of time. We found that the variances were comparable initially, but as time progressed the variance of the bootstrap filter estimate grew markedly in comparison to the variance of the series expansion filter.
Figure 2: Plot of the position components from a sample path of the coordinated turn model, with the mean position as estimated by the SE-UPF (black dots).

In Figure 4, we show normalised histograms that display the root mean squared error from each run of both filters. The error in the position and velocity of the SE-UPF estimate is noticeably lower than that of the bootstrap filter.

6 Discussion

Our experiments suggest that the SE-UPF is more robust than the bootstrap filter. In some scenarios it also outperforms the random weight particle filter, while being more generally applicable. The methodology of section 3 suggests some interesting possibilities. It may be possible to use heavy-tailed importance distributions in order to provide theoretical guarantees on the variance of the importance weights.

Given the improvement of the SE-UPF over the bootstrap filter, it would be interesting to investigate the application of series expansion importance sampling to particle Markov chain Monte-Carlo (PMCMC) methods. The estimation of parameters of a diffusion process is a difficult task in general, but PMCMC methods are a promising tool for this problem. However, the bootstrap filter can struggle to cope within PMCMC. The filter will be run repeatedly, and the particles will often use dynamics that are very different from the dynamics that generated the observations. As a result, most observations will behave like outliers, and the effective sample size of the filter will be lower.

The SE-UPF can direct particles to ‘interesting’ areas, and has been shown to have a lower variance than the bootstrap filter. For this reason, we expect that it will improve the performance of a PMCMC algorithm.
Figure 3: Relative performance of bootstrap and series expansion filters for the coordinated turn model experiment.

(a) Relative variance of \( \mathbb{E}[X_1^{(1)}|Y_{1:t}] \)
(b) Relative variance of \( \mathbb{E}[X_1^{(2)}|Y_{1:t}] \)
(c) Relative variance of \( \mathbb{E}[X_1^{(3)}|Y_{1:t}] \)
(d) Relative variance of \( \mathbb{E}[X_1^{(4)}|Y_{1:t}] \)

Figure 4: RMSE for position, velocity, and turn rate estimates from both filters. The SE-UPF error (broken line) tends to be smaller than the bootstrap error (solid line).

(a) Position errors
(b) Velocity errors
(c) Turn rate errors
References


