

Nonlinear filtering via stochastic PDE projection on mixture manifolds in L^2 direct metric

John Armstrong¹ · Damiano Brigo²

Received: 20 August 2014 / Accepted: 11 November 2015

© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract We examine some differential geometric approaches to finding approximate solutions to the continuous time nonlinear filtering problem. Our primary focus is a new projection method for the optimal filter infinite-dimensional stochastic partial differential equation (SPDE), based on the direct L^2 metric and on a family of normal mixtures. This results in a new finite-dimensional approximate filter based on the differential geometric approach to statistics. We compare this new filter to earlier projection methods based on the Hellinger distance/Fisher metric and exponential families, and compare the L^2 mixture projection filter with a particle method with the same number of parameters, using the Levy metric. We discuss differences between projecting the SPDE for the normalized density, known as Kushner–Stratonovich equation, and the SPDE for the unnormalized density known as Zakai equation. We prove that for a simple choice of the mixture manifold the L^2 mixture projection filter coincides with a Galerkin method, whereas for more general mixture manifolds the equivalence does not hold and the L^2 mixture filter is more general. We study particular systems that may illustrate the advantages of this new filter over other algorithms when comparing outputs with the optimal filter. We finally consider a specific software design that is suited for a numerically efficient implementation of this filter and provide numerical examples. We leverage an algebraic ring structure by proving that in presence of a given structure in the system coefficients the key integrations needed to implement the new filter equations can be executed offline.

✉ Damiano Brigo
damiano.brigo@imperial.ac.uk

John Armstrong
john.l.armstrong@kcl.ac.uk

¹ Department of Mathematics, King's College, London, UK

² Department of Mathematics, Imperial College, London, UK

Keywords Direct L^2 metric · Exponential families · Finite-dimensional families of probability distributions · Fisher information · Hellinger distance · Levy metric · Mixture families · Stochastic filtering · Galerkin methods

1 Introduction

In the nonlinear filtering problem, one observes a system whose state is known to follow a given stochastic differential equation. The observations that have been made contain an additional noise term, so one cannot hope to know the true state of the system. However, one can reasonably ask what is the probability density over the possible states conditional on the given observations. When the observations are made in continuous time, the probability density follows a stochastic partial differential equation known as the Kushner–Stratonovich equation. This can be seen as a generalization of the Fokker–Planck equation that expresses the evolution of the density of a diffusion process. Thus, the problem we wish to address boils down to finding approximate solutions to the Kushner–Stratonovich equation. The literature on stochastic filtering for nonlinear systems is vast and it is impossible to do justice to all past contributions here. For a proof of the fact that the solution is infinite dimensional see, for example, Hazewinkel et al. [26] for the cubic sensor, whereas for special nonlinear systems still admitting an exact finite-dimensional filter see, for example, Beneš [14] and Beneš and Elliott [15]. For a more complete treatment of the filtering problem from a mathematical point of view see for example Lipster and Shiryayev [37]. See Jazwinski [29] for a more applied perspective and comprehensive insight.

The main idea we will employ is inspired by the differential geometric approach to statistics developed in [5] and [44]. The idea of applying this approach to the filtering problem has been sketched first in [25]. One thinks of the probability distribution as evolving in an infinite-dimensional space \mathcal{P} which is in turn contained in some Hilbert space H . One can then think of the Kushner–Stratonovich equation as defining a vector field in \mathcal{P} : the integral curves of the vector field should correspond to the solutions of the equation. To find approximate solutions to the Kushner–Stratonovich equation, one chooses a finite-dimensional submanifold M of H and approximates the probability distributions as points in M . At each point of M , one can use the Hilbert space structure to project the vector field onto the tangent space of M . One can now attempt to find approximate solutions to the Kushner–Stratonovich equations by integrating this vector field on the manifold M . This mental image is slightly inaccurate. The Kushner–Stratonovich equation is a stochastic PDE rather than a PDE so one should imagine some kind of stochastic vector field rather than a smooth vector field. Thus, in this approach we hope to approximate the infinite-dimensional stochastic PDE by solving a finite-dimensional stochastic ODE on the manifold. Note that our approximation will depend upon two choices: the choice of manifold M and the choice of Hilbert space structure H . In this paper, we will consider two possible choices for the Hilbert space structure: the direct L^2 metric on the space of probability distributions; the Hilbert space structure associated with the Hellinger distance and the Fisher Information metric. Our focus will be on the direct L^2 metric since projection using the Hellinger distance has been considered before. As we shall see, the choice of the “best” Hilbert space structure is determined by the manifold one wishes to consider—for manifolds

associated with exponential families of distributions the Hellinger metric leads to the simplest equations, whereas the direct L^2 metric works well with mixture distributions. It was proven in [18] that the projection filter in Hellinger metric on exponential families is equivalent to the classical assumed density filters. In this paper, we show that the projection filter for basic mixture manifolds in L^2 metric is equivalent to a Galerkin method. This only holds for very basic mixture families, however, and the L^2 projection method turns out to be more general than the Galerkin method. We will write down the stochastic ODE determined by the geometric approach when $H = L^2$ and show how it leads to a numerical scheme for finding approximate solutions to the Kushner–Stratonovich equations in terms of a mixture of normal distributions. We will call this scheme the L^2 normal mixture projection filter or simply the L2NM projection filter. The stochastic ODE for the Hellinger metric was considered in [17, 19] and [18]. In particular, a precise numerical scheme is given in [19] for finding solutions by projecting onto an exponential family of distributions. We will call this scheme the Hellinger exponential projection filter or simply the HE projection filter. We will compare the results of a C++ implementation of the L2NM projection filter with a number of other numerical approaches including the HE projection filter and the optimal filter. We can measure the goodness of our filtering approximations thanks to the geometric structure, and in particular, the precise metrics we are using on the spaces of probability measures. What emerges is that the two projection methods produce excellent results for a variety of filtering problems. The results appear similar for both projection methods, which gives more accurate results depends upon the problem. As we shall see, however, the L2NM projection approach can be implemented more efficiently. In particular, one needs to perform numerical integration as part of the HE projection filter algorithm whereas all integrals that occur in the L2NM projection can be evaluated analytically. We also compare the L2NM filter to a particle filter with the best possible combination of particles with respect to the Lévy metric. Introducing the Lévy metric is needed because particles' densities do not compare well with smooth densities when using L^2 induced metrics. We show that, given the same number of parameters, the L2NM may outperform a particle-based system. We should also mention that the systems we analyse here are one-dimensional, and that we plan to address large-dimensional systems in a subsequent paper.

The paper is structured as follows: In Sect. 2, we introduce the nonlinear filtering problem and the infinite-dimensional Stochastic PDE (SPDE) that solves it. In Sect. 3, we introduce the geometric structure we need to project the filtering SPDE onto a finite-dimensional manifold of probability densities. In Sect. 4, we perform the projection of the filtering SPDE according to the L2NM framework and also recall the HE based framework. In Sect. 5, we prove equivalence between the projection filter in L^2 metric for basic mixture families and the Galerkin method. In Sect. 6, we briefly introduce the main issues in the numerical implementation and then focus on software design for the L2NM filter. In Sect. 7, a second theoretical result is provided, showing a particularly convenient structure for the projection filter equations for specific choices of the system properties and of the mixture manifold. In Sect. 8, we look at numerical results, whereas in Sect. 9 we compare our outputs with a particle method. In Sect. 10, we compare our filters with a robust implementation of the optimal filter based on Hermite functions. Section 11 concludes the paper.

2 The nonlinear filtering problem with continuous time observations

In the nonlinear filtering problem, the state of some system is modelled by a process X called the signal. This signal evolves over time t according to an Itô stochastic differential equation (SDE). We measure the state of the system using some observation Y . The observations are not accurate, there is a noise term. So the observation Y is related to the signal X by a second equation.

$$dX_t = f_t(X_t) dt + \sigma_t(X_t) dW_t, \quad X_0, \quad dY_t = b_t(X_t) dt + dV_t, \quad Y_0 = 0. \quad (1)$$

In these equations the unobserved state process $\{X_t, t \geq 0\}$ takes values in \mathbb{R}^n , the observation $\{Y_t, t \geq 0\}$ takes values in \mathbb{R}^d and the noise processes $\{W_t, t \geq 0\}$ and $\{V_t, t \geq 0\}$ are two Brownian motions. The nonlinear filtering problem consists in finding the conditional probability distribution π_t of the state X_t given the observations up to time t and the prior distribution π_0 for X_0 . Let us assume that X_0 , and the two Brownian motions are independent. Let us also assume that the covariance matrix for V_t is invertible. We can then assume without any further loss of generality that its covariance matrix is the identity. We introduce a variable a_t defined by $a_t = \sigma_t \sigma_t^T$. With these preliminaries, and a number of rather more technical conditions which we will state shortly, one can show that π_t satisfies the Kushner–Stratonovich equation. This states that for any compactly supported test function ϕ defined on \mathbb{R}^n

$$\begin{aligned} \pi_t(\phi) = \pi_0(\phi) &+ \int_0^t \pi_s(\mathcal{L}_s \phi) ds \\ &+ \sum_{k=1}^d \int_0^t [\pi_s(b_s^k \phi) - \pi_s(b_s^k) \pi_s(\phi)] [dY_s^k - \pi_s(b_s^k) ds], \end{aligned} \quad (2)$$

where for all $t \geq 0$, the backward diffusion operator \mathcal{L}_t and its formal adjoint \mathcal{L}^* are defined by

$$\begin{aligned} \mathcal{L}_t = \sum_{i=1}^n f_t^i \frac{\partial}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^n a_t^{ij} \frac{\partial^2}{\partial x_i \partial x_j}, \quad \mathcal{L}_t^* \phi = - \sum_{i=1}^n \frac{\partial}{\partial x_i} [f_t^i \phi] \\ + \frac{1}{2} \sum_{i,j=1}^n \frac{\partial^2}{\partial x_i \partial x_j} [a_t^{ij} \phi]. \end{aligned}$$

We assume now that π_t can be represented by a density p_t with respect to the Lebesgue measure on \mathbb{R}^n for all time $t \geq 0$ and that we can replace the term involving $\mathcal{L}_s \phi$ with a term involving its formal adjoint \mathcal{L}^* . Thus, proceeding formally, we find that p_t obeys the following Itô-type stochastic partial differential equation (SPDE):

$$dp_t = \mathcal{L}_t^* p_t dt + \sum_{k=1}^d p_t [b_t^k - E_{p_t}\{b_t^k\}] [dY_t^k - E_{p_t}\{b_t^k\} dt] \quad (3)$$

where $E_{p_t}\{\cdot\}$ denotes the expectation with respect to the probability density p_t (equivalently the conditional expectation given the observations up to time t). This equation is written in Itô form. When working with stochastic calculus on manifolds it is customary to use Fisk–Stratonovich–McShane SDE’s (Stratonovich from now on) rather than Itô SDE’s. This is because if one bases the analysis of the SDE on the drift and diffusion coefficient vector fields, then the SDE coefficients change covariantly in Stratonovich calculus, whereas the second-order terms arising in Ito calculus break the covariant nature of the coordinate transformation. However, we will discuss in further work how one may work directly with Itô SDEs on manifolds and how this may relate to different notions of projection than the Stratonovich vector field projection adopted here and in the previous projection filter papers. This will relate importantly to the projection optimality.

A straightforward calculation based on the Itô–Stratonovich transformation formula yields the following Stratonovich calculus version of the Itô-calculus Kushner–Stratonovich SPDE (3):

$$dp_t = \mathcal{L}_t^* p_t dt - \frac{1}{2} p_t [|b_t|^2 - E_{p_t}\{|b_t|^2\}] dt + \sum_{k=1}^d p_t [b_t^k - E_{p_t}\{b_t^k\}] \circ dY_t^k.$$

We have indicated that this is the Stratonovich form of the equation by the presence of the symbol ‘ \circ ’ inbetween the diffusion coefficient and the Brownian motion of the SDE. We shall use this convention throughout the rest of the paper. To simplify notation, we introduce the following definitions:

$$\gamma_t^0(p) := \frac{1}{2} [|b_t|^2 - E_p\{|b_t|^2\}] p, \quad \gamma_t^k(p) := [b_t^k - E_p\{b_t^k\}] p, \tag{4}$$

for $k = 1, \dots, d$. The Stratonovich form of the Kushner–Stratonovich equation reads now

$$dp_t = \mathcal{L}_t^* p_t dt - \gamma_t^0(p_t) dt + \sum_{k=1}^d \gamma_t^k(p_t) \circ dY_t^k. \tag{5}$$

Thus, subject to the assumption that a density p_t exists for all time and assuming the necessary decay condition to ensure that replacing \mathcal{L} with its formal adjoint is valid, we find that solving the nonlinear filtering problem is equivalent to solving this SPDE. Numerically approximating the solution of Eq. (5) is the primary focus of this paper. The technical conditions required in order for Eq. (2) to follow from (1) are known in the filtering literature and involve local Lipschitz continuity for f and a , linear growth in $|x|^2$ for $x^T f_t(x)$ and trace $a_t(x)$, and polynomial growth for $|b_t(x)|$ in $|x|$. See [6] for the details in the same notation and references therein, especially Fujisaki, Kallianpur and Kunita [30].

We should point out that, in a broad part of the nonlinear filtering literature, the preferred SPDE for the optimal filter is an SPDE for an unnormalized version q of the optimal filter density p . The Duncan–Mortensen–Zakai equation (Zakai for brevity) for the unnormalized density $q_t(x)$ of the optimal filter reads, in Stratonovich form

$$dq_t = \mathcal{L}_t^* q_t dt - \frac{1}{2} q_t |b_t|^2 dt + \sum_{k=1}^d q_t [b_t^k] \circ dY_t^k, \quad q_0 = p_0,$$

see, for example, Eq. 14.31 in [2], where conditions under which this is an evolution equation in L^2 are discussed. This is a linear Stochastic PDE and as such it is more tractable than the KS equation. Linearity has led the Zakai Eq. to be preferred to the KS Eq. in general. The reason why we still resort to KS will be clarified in full detail when we derive the projection filter below, although we may anticipate that this has to do with the fact that we aim to derive an approximation that is good for the normalized density p and not for the unnormalized density q . There is a second and related reason why the Zakai Eq. has been preferred to the KS Eq. in the literature: the possibility to derive a robust PDE for the optimal filter, which we briefly review now.

While here we use Stratonovich calculus to deal with manifolds, nonlinear filtering equations are usually based on Ito stochastic differential equations, holding almost surely in the space of continuous functions. However, in practice real-world sample paths have finite variation, and the set of finite variation paths has measure zero under the Wiener measure. It is in principle possible to obtain a version of the filter which takes arbitrary values on any real-world sample path. In technical terms, nonlinear filtering equations such as the KS or Zakai equation are not robust. One would wish to have that the nonlinear filter is a continuous functional on the continuous functions (paths). This way we would have that the filter based on real-life finite variation paths is close to the theoretical optimal one based on unbounded variation paths. There are essentially two approaches to bypass the above lack of robustness. In one approach, introduced by Clark [21], the Zakai stochastic PDE is transformed into an equivalent pathwise form avoiding the differential dY_t of the observation process. Indeed, assuming the observation function b to be time homogeneous for simplicity, $b_t(x) = b(x)$, set

$$r_t(x) := \exp\left(-\sum_{k=1}^d b^k(x) Y_t^k\right) q_t(x).$$

Then, by the standard chain rule of Stratonovich calculus one derives easily

$$\partial_t r_t = \frac{\mathcal{L}_t^* \left(\exp\left(\sum_{k=1}^d b^k Y_t^k\right) r_t \right)}{\exp\left(\sum_{k=1}^d b^k Y_t^k\right)} - \frac{1}{2} |b|^2 r_t.$$

This transformed filtering equation is shown to be continuous with respect to the observation under a suitable topology. This equation can then be extended to real-life sample paths using continuity [22].

In the other approach to robust filtering, due originally to Balakrishnan [8], one tries to model the observation process directly with a white noise error term. Although modelling white noise directly is intuitively appealing, it brings a host of mathematical complications. Kallianpur and Karandikar [31] developed the theory of nonlinear

filtering in this framework, while [7] extends this second approach to the case of correlated state and observation noises.

Going back to the projection filter, one could consider using a projection method to find approximate solutions for r_t . However, as our primary interest is in finding good approximations to p_t rather than r_t , we believe that projecting the KS equation is the more promising approach. For comparison purposes, we have implemented below the robust optimal filter using the method of Luo and Yau [38–40], who put emphasis on real-time implementation.

3 Statistical manifolds

3.1 Families of distributions

As discussed in the introduction, the idea of a projection filter is to approximate solutions to the Kushner–Stratonovich Eq. (2) using a finite-dimensional family of distributions.

Example 3.1 A normal mixture family contains distributions given by:

$$p = \sum_{i=1}^m \lambda_i \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(\frac{-(x - \mu_i)^2}{2\sigma_i^2}\right)$$

with $\lambda_i > 0$ and $\sum \lambda_i = 1$. It is a $3m - 1$ -dimensional family of distributions.

Example 3.2 A polynomial exponential family contains distributions given by:

$$p = \exp\left(\sum_{i=0}^m a_i x^i\right)$$

where a_0 is chosen to ensure that the integral of p is equal to 1. To ensure the convergence of the integral we must have that m is even and $a_m < 0$. This is an m -dimensional family of distributions. Polynomial exponential families are a special case of the more general notion of an exponential family, see for example [5].

A key motivation for considering these families is that one can reproduce many of the qualitative features of distributions that arise in practice using these distributions. For example, consider the qualitative specification: the distribution should be bimodal with peaks near -1 and 1 with the peak at -1 twice as high and twice as wide as the peak near 1 . One can easily write down a distribution of this approximate form using a normal mixture. To find a similar exponential family, one seeks a polynomial with: local maxima at -1 and 1 ; with the maximum values at these points differing by $\log(2)$; with second derivative at 1 equal to twice that at -1 . These conditions give linear equations in the polynomial coefficients. Using degree 6 polynomials it is simple to find solutions meeting all these requirements. A specific numerical example of a polynomial meeting these requirements is given in the exponent of the exponential distribution plotted in

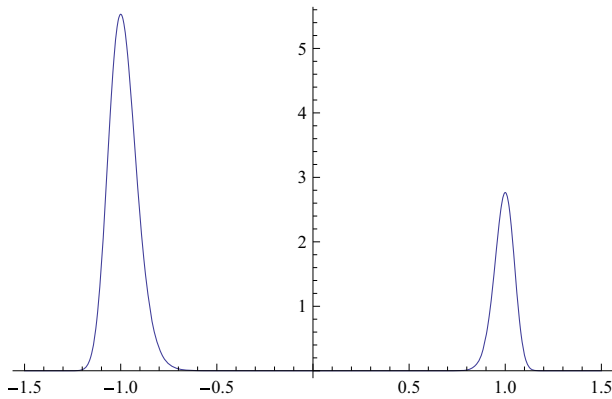


Fig. 1 $y = \exp(-18.98 - 13.15x + 23.54x^2 + 25.43x^3 + 13.96x^4 - 12.63x^5 - 17.15x^6)$

Fig. 1, see [6] for more details. We see that normal mixtures and exponential families have a broadly similar power to describe the qualitative shape of a distribution using only a small number of parameters. Our hope is that by approximating the probability distributions that occur in the Kushner–Stratonovich equation by elements of one of these families we will be able to derive a low-dimensional approximation to the full infinite-dimensional stochastic partial differential equation.

Mixture families have a long tradition in filtering. Alspach and Sorenson [4] already highlight that Gaussian sums work well in nonlinear filtering. Ahmed [2] points out that Gaussian densities are dense in L^2 , pointing at the fact that with a sufficiently large number of components a mixture of Gaussian densities may be able to reproduce most features of squared integrable densities.

3.2 Two Hilbert spaces of probability distributions

We have given direct parameterisations of our families of probability distributions and thus have implicitly represented them as finite-dimensional manifolds. In this section, we will see how families of probability distributions can be thought of as being embedded in a Hilbert space and hence they inherit a manifold structure and metric from this Hilbert space. There are two obvious ways of thinking of embedding a probability density function on \mathbb{R}^n in a Hilbert space. The first is to simply assume that the probability density function is square integrable and hence lies directly in $L^2(\mathbb{R}^n)$. The second is to use the fact that a probability density function lies in $L^1(\mathbb{R}^n)$ and is non-negative almost everywhere. Hence \sqrt{p} will lie in $L^2(\mathbb{R}^n)$. For clarity we will write $L^2_D(\mathbb{R}^n)$ when we think of $L^2(\mathbb{R}^n)$ as containing densities directly. The D stands for direct. We write $\mathcal{D} \subset L^2_D(\mathbb{R}^n)$ where \mathcal{D} is the set of square integrable probability densities (functions with integral 1 which are positive almost everywhere). Similarly we will write $L^2_H(\mathbb{R}^n)$ when we think of $L^2(\mathbb{R}^n)$ as being a space of square roots of densities. The H stands for Hellinger (for reasons we will explain shortly). We will write \mathcal{H} for the subset of $L^2_H(\mathbb{R}^n)$ consisting of square roots of probability densities. We now have two possible ways of formalizing the notion of a family of probability

distributions. In the next section, we will define a smooth family of distributions to be either a smooth submanifold of L^2_D which also lies in \mathcal{D} or a smooth submanifold of L^2_H which also lies in \mathcal{H} . Either way the families we discussed earlier will give us finite-dimensional families in this more formal sense. The Hilbert space structures of L^2_D and L^2_H allow us to define two notions of distance between probability distributions which we will denote d_D and d_H . Given two probability distributions p_1 and p_2 , we have an injection ι into L^2 so one defines the distance to be the norm of $\iota(p_1) - \iota(p_2)$. So given two probability densities p_1 and p_2 on \mathbb{R}^n we can define:

$$d_H(p_1, p_2) = \left(\int (\sqrt{p_1} - \sqrt{p_2})^2 d\mu \right)^{\frac{1}{2}}, \quad d_D(p_1, p_2) = \left(\int (p_1 - p_2)^2 d\mu \right)^{\frac{1}{2}}.$$

Here, $d\mu$ is the Lebesgue measure. d_H defines the Hellinger distance between the two distributions, which explains our use of H as a subscript. We will write $\langle \cdot, \cdot \rangle_H$ for the inner product associated with d_H and $\langle \cdot, \cdot \rangle_D$ or simply $\langle \cdot, \cdot \rangle$ for the inner product associated with d_D . In this paper, we will consider the projection of the conditional density of the true state of the system given the observations (which is assumed to lie in \mathcal{D} or \mathcal{H}) onto a submanifold. The notion of projection only makes sense with respect to a particular inner product structure. Thus, we can consider projection using d_H or projection using d_D . Each has advantages and disadvantages. The most notable advantage of the Hellinger metric is that the d_H metric can be defined independently of the Lebesgue measure and its definition can be extended to define the distance between measures without density functions (see Jacod and Shiryaev [28]). In particular, the Hellinger distance is independent of the choice of parameterization for \mathbb{R}^n . This is a very attractive feature in terms of the differential geometry of our setup. Despite the significant theoretical advantages of the d_H metric, the d_D metric has an obvious advantage when studying mixture families: it comes from an inner product on L^2_D and so commutes with addition on L^2_D . So it should be relatively easy to calculate with the d_D metric when adding distributions as happens in mixture families. As we shall see in practice, when one performs concrete calculations, the d_H metric works well for exponential families and the d_D metric works well for mixture families.

3.3 The tangent space of a family of distributions

To make our notion of smooth families precise, we need to explain what we mean by a smooth map into an infinite-dimensional space. Let U and V be Hilbert spaces and let $f : U \rightarrow V$ be a continuous map (f need only be defined on some open subset of U). We say that f is Fréchet differentiable at x if there exists a bounded linear map $A : U \rightarrow V$ satisfying:

$$\lim_{h \rightarrow x} \frac{\|f(h) - f(x) - Ah\|_V}{\|h\|_U} = 0$$

If A exists it is unique and we denote it by $Df(x)$. This limit is called the Fréchet derivative of f at x . It is the best linear approximation to f at 0 in the sense of

minimizing the norm on V . This allows us to define a smooth map $f : U \rightarrow V$ defined on an open subset of U to be an infinitely Fréchet differentiable map. We define an immersion of an open subset of \mathbb{R}^n into V to be a map such that $Df(x)$ is injective at every point where f is defined. The latter condition ensures that the best linear approximation to f is a genuinely n -dimensional map. Given an immersion f defined on a neighbourhood of x , we can think of the vector subspace of V given by the image of $Df(x)$ as representing the tangent space at x . To make these ideas more concrete, let us suppose that $p(\theta)$ is a probability distribution depending smoothly on some parameter $\theta = (\theta_1, \theta_2, \dots, \theta_m) \in U$ where U is some open subset of \mathbb{R}^m . The map $\theta \rightarrow p(\theta)$ defines a map $i : U \rightarrow \mathcal{D}$. At a given point $\theta \in U$ and for a vector $h = (h_1, h_2, \dots, h_m) \in \mathbb{R}^m$, we can compute the Fréchet derivative to obtain:

$$Di(\theta)h = \sum_{i=1}^m \frac{\partial p}{\partial \theta_i} h_i$$

So we can identify the tangent space at θ with the following subspace of $L^2_{\mathcal{D}}$:

$$\text{span} \left\{ \frac{\partial p}{\partial \theta_1}, \frac{\partial p}{\partial \theta_2}, \dots, \frac{\partial p}{\partial \theta_m} \right\} \tag{6}$$

We can formally define a smooth n -dimensional family of probability distributions in $L^2_{\mathcal{D}}$ to be an immersion of an open subset of \mathbb{R}^n into \mathcal{D} . Equivalently, it is a smoothly parameterized probability distribution p such that the above vectors in L^2 are linearly independent. We can define a smooth m -dimensional family of probability distributions in L^2_H in the same way. This time let $q(\theta)$ be a square root of a probability distribution depending smoothly on θ . The tangent vectors in this case will be the partial derivatives of q with respect to θ . Since one normally prefers to work in terms of probability distributions rather than their square roots, we use the chain rule to write the tangent space as:

$$\text{span} \left\{ \frac{1}{2\sqrt{p}} \frac{\partial p}{\partial \theta_1}, \frac{1}{2\sqrt{p}} \frac{\partial p}{\partial \theta_2}, \dots, \frac{1}{2\sqrt{p}} \frac{\partial p}{\partial \theta_m} \right\} \tag{7}$$

We have defined a family of distributions in terms of a single immersion f into a Hilbert space V . In other words, we have defined a family of distributions in terms of a specific parameterization of the image of f . It is tempting to try and phrase the theory in terms of the image of f . To this end, one defines an embedded submanifold of V to be a subspace of V which is covered by immersions f_i from open subsets of \mathbb{R}^n where each f_i is a homeomorphisms onto its image. With this definition, we can state that the tangent space of an embedded submanifold is independent of the choice of parameterization.

One might be tempted to talk about submanifolds of the space of probability distributions, but one should be careful. The spaces \mathcal{H} and \mathcal{D} are not open subsets of L^2_H and $L^2_{\mathcal{D}}$ and so do not have any obvious Hilbert-manifold structure. To see why, one may perturb a positive density by a negative spike with an arbitrarily small area

and obtain a function arbitrarily close to the density in L^2 norm but not almost surely positive, see [6] for a graphic illustration.

3.4 The Fisher information metric

Given two tangent vectors at a point to a family of probability distributions, we can form their inner product using $\langle \cdot, \cdot \rangle_H$. This defines a so-called Riemannian metric on the family. With respect to a particular parameterization θ , we can compute the inner product of the i th and j th basis vectors given in Eq. (7). We call this quantity $\frac{1}{4}g_{ij}$.

$$\begin{aligned} \frac{1}{4}g_{ij}(\theta) &:= \left\langle \frac{1}{2\sqrt{p}} \frac{\partial p}{\partial \theta_i}, \frac{1}{2\sqrt{p}} \frac{\partial p}{\partial \theta_j} \right\rangle_H = \frac{1}{4} \int \frac{1}{p} \frac{\partial p}{\partial \theta_i} \frac{\partial p}{\partial \theta_j} d\mu \\ &= \frac{1}{4} \int \frac{\partial \log p}{\partial \theta_i} \frac{\partial \log p}{\partial \theta_j} p d\mu = \frac{1}{4} E_p \left(\frac{\partial \log p}{\partial \theta_i} \frac{\partial \log p}{\partial \theta_j} \right) \end{aligned}$$

Up to the factor of $\frac{1}{4}$, this last formula is the standard definition for the Fisher information matrix. So our g_{ij} is the Fisher information matrix. We can now interpret this matrix as the Fisher information metric and observe that, up to the constant factor, this is the same thing as the Hellinger distance. See [5] and [1] for more in depth study on this differential geometric approach to statistics.

Example 3.3 The Gaussian family of densities can be parameterized using parameters mean μ and variance v . With this parameterization the Fisher metric is given by:

$$g(\mu, v) = \frac{1}{v} \begin{bmatrix} 1 & 0 \\ 0 & 1/(2v) \end{bmatrix}$$

The Gaussian family may also be considered with different coordinates, namely as a particular exponential family, see [6] for the metric g under these coordinates.

The particular importance of the metric structure for this paper is that it allows us to define orthogonal projection of L^2_H onto the tangent space. Suppose that one has m linearly independent vectors w_i spanning some subspace W of a Hilbert space V . By linearity, one can write the orthogonal projection onto W as:

$$\Pi(v) = \sum_{i=1}^m \left[\sum_{j=1}^m A^{ij} \langle v, w_j \rangle \right] w_i$$

for some appropriately chosen constants A^{ij} . Since Π acts as the identity on w_i , we see that A^{ij} must be the inverse of the matrix $A_{ij} = \langle w_i, w_j \rangle$. We can apply this to the basis given in Eq. (7). Defining g^{ij} to be the inverse of the matrix g_{ij} we obtain the following formula for projection, using the Hellinger metric, onto the tangent space of a family of distributions:

$$\Pi_H(v) = \sum_{i=1}^m \left[\sum_{j=1}^m 4g^{ij} \left\langle v, \frac{1}{2\sqrt{p}} \frac{\partial p}{\partial \theta_j} \right\rangle_H \right] \frac{1}{2\sqrt{p}} \frac{\partial p}{\partial \theta_i}.$$

3.5 The direct L^2 metric

The ideas from the previous section can also be applied to the direct L^2 metric. This gives a different Riemannian metric on the manifold. We will write $h = h_{ij}$ to denote the L^2 metric when written with respect to a particular parameterization.

Example 3.4 In coordinates μ, v , the L^2 metric on the Gaussian family is (compare with Example 3.3):

$$h(\mu, v) = \frac{1}{4v\sqrt{v\pi}} \begin{bmatrix} 1 & 0 \\ 0 & \frac{3}{8v} \end{bmatrix}$$

We can obtain a formula for projection in L^2_D using the direct L^2 metric using the basis given in Eq. (6). We write h^{ij} for the matrix inverse of h_{ij} .

$$\Pi_D(v) = \sum_{i=1}^m \left[\sum_{j=1}^m h^{ij} \left\langle v, \frac{\partial p}{\partial \theta_j} \right\rangle_D \right] \frac{\partial p}{\partial \theta_i}. \tag{8}$$

4 The projection filter

Given a family of probability distributions parameterised by θ , we wish to approximate an infinite-dimensional solution to the nonlinear filtering SPDE using elements of this family. Thus, we take the Kushner–Stratonovich Eq. (5), view it as defining a stochastic vector field in \mathcal{D} and then project that vector field onto the tangent space of our family. The projected equations can then be viewed as giving a stochastic differential equation for θ . In this section, we will write down these projected equations explicitly. Let $\theta \rightarrow p(\theta)$ be the parameterization for our family. A curve $t \rightarrow \theta(t)$ in the parameter space corresponds to a curve $t \rightarrow p(\cdot, \theta(t))$ in \mathcal{D} . For such a curve, the left-hand side of the Kushner–Stratonovich Eq. (5) can be written:

$$d_t p(\cdot, \theta(t)) = \sum_{i=1}^m \frac{\partial p(\cdot, \theta(t))}{\partial \theta_i} d_t \theta_i(t) = \sum_{i=1}^m v_i d\theta_i$$

where we write $v_i = \frac{\partial p}{\partial \theta_i}$. $\{v_i\}$ is the basis for the tangent space of the manifold at $\theta(t)$. Given the projection formula given in Eq. (8), we can project the terms on the right-hand side onto the tangent space of the manifold using the direct L^2 metric as follows:

$$\begin{aligned} \Pi_D^\theta[\mathcal{L}^* p] &= \sum_{i=1}^m \left[\sum_{j=1}^m h^{ij} \langle \mathcal{L}^* p, v_j \rangle \right] v_i = \sum_{i=1}^m \left[\sum_{j=1}^m h^{ij} \langle p, \mathcal{L} v_j \rangle \right] v_i \\ \Pi_D^\theta[\gamma^k(p)] &= \sum_{i=1}^m \left[\sum_{j=1}^m h^{ij} \langle \gamma^k(p), v_j \rangle \right] v_i \end{aligned}$$

Thus, if we take L^2 projection of each side of Eq. (5) we obtain:

$$\sum_{i=1}^m v_i d\theta^i = \sum_{i=1}^m \left[\sum_{j=1}^m h^{ij} \left\{ \langle p, \mathcal{L} v_j \rangle dt - \langle \gamma^0(p), v_j \rangle dt + \sum_{k=1}^d \langle \gamma^k(p), v_j \rangle \circ dY^k \right\} \right] v_i$$

Since the v_i form a basis of the tangent space, we can equate the coefficients of v_i to obtain the following

Proposition 4.1 (New filter: The d_D KS projection filter) *Projecting in d_D metric the KS SPDE for the optimal filter onto the statistical manifold $p(\cdot, \theta)$ leads to the d_D -based projection filter equation*

$$d\theta^i = \sum_{j=1}^m h^{ij} \left\{ \langle p(\theta), \mathcal{L} v_j \rangle dt - \langle \gamma^0(p(\theta)), v_j \rangle dt + \sum_{k=1}^d \langle \gamma^k(p(\theta)), v_j \rangle \circ dY^k \right\}. \tag{9}$$

This is the promised finite-dimensional stochastic differential equation for θ corresponding to L^2 projection.

If preferred, one could project the Kushner–Stratonovich equation using the Hellinger metric instead. This yields the following stochastic differential equation derived originally in [19]:

Proposition 4.2 *Projecting in Hellinger metric the KS SPDE for the optimal filter onto the statistical manifold $p(\cdot, \theta)$ leads to the Hellinger-based projection filter equation*

$$d\theta_i = \sum_{j=1}^m g^{ij} \left(\left\langle \frac{\mathcal{L}^* p(\theta)}{p(\theta)}, v_j \right\rangle dt - \left\langle \frac{1}{2} |b|^2, v_j \right\rangle dt + \sum_{k=1}^d \langle b^k, v_j \rangle \circ dY^k \right) \tag{10}$$

Note that the inner products in this equation are the direct L^2 inner products: we are simply using the L^2 inner product notation as a compact notation for integrals.

It is now possible to explain why we resorted to the KS Equation for the optimal filter rather than the Zakai equation in deriving the projection filter.

Consider the nonlinear terms in the KS Eq. (5), namely

$$\frac{1}{2} p_t E_{p_t} \{|b_t|^2\} dt, \quad \sum_{k=1}^d p_t [-E_{p_t} \{b_t^k\}] \circ dY_t^k.$$

Consider first the Hellinger projection filter (10). By inspection, we see that there is no impact of the nonlinear terms in the projected equation. Therefore, we have the following.

Proposition 4.3 *Projecting the Zakai unnormalized density SPDE in Hellinger metric onto the statistical manifold $p(\cdot, \theta)$ results in the same projection filter equation as projecting the normalized density KS SPDE.*

The main focus of this paper, however, is the d_D projection filter (9). For this filter, we do have an impact of the nonlinear terms. In fact, it is easy to adapt the derivation of the d_D filter to the Zakai equation, which leads to the following new filter.

Proposition 4.4 (New filter: The d_D Zakai projection filter) *Projection in d_D metric of the Zakai SPDE for a specific unnormalized version of the optimal filter density results in the following projection filter:*

$$d\theta^i = \sum_{j=1}^m h^{ij} \left\{ \langle p(\theta), \mathcal{L}v_j \rangle dt - \langle \frac{1}{2} |b_t|^2 p(\theta), v_j \rangle dt + \sum_{k=1}^d \langle b_t^k p(\theta), v_j \rangle \circ dY^k \right\}. \quad (11)$$

This filter is clearly different from (9). We have implemented (11) for the sensor case, using a simple variation on the numerical algorithms below. We found that (11) gives slightly worse results for the L^2 residual of the normalized density than (9). This can be explained simply by the fact that if we want to approximate p in a given norm then we should project an equation for p whereas if we wish to approximate q we should project an equation for q . The fact that \sqrt{q} has variable L^2 norm in time is not relevant for the projection in the Hellinger metric, while it is for the d_D metric, where the lack of normalization in the Zakai Eq. density plays a role in the projection.

5 Equivalence with assumed density filters and Galerkin methods

The projection filter with specific metrics and manifolds can be shown to be equivalent to earlier filtering algorithms. In particular, while the d_H metric leads to the Fisher Information and to an equivalence between the projection filter and assumed density filters (ADFs) when using exponential families, see [18], the d_D metric for simple mixture families is equivalent to a Galerkin method, as we show now following the second named author preprint [16]. For applications of Galerkin methods to nonlinear filtering, we refer for example to [9,24,27,41]. Ahmed [2], Chapter 14, Sections 14.3 and 14.4, summarizes the Galerkin method for the Zakai equation, see also [3]. Pardoux [43] uses Galerkin techniques to analyse existence of solutions for the nonlinear filtering SPDE. Bensoussan et al. [10] adopt the splitting up method. We also refer to Frey et al [23] for Galerkin methods applied to the extended Zakai equation for diffusive and point process observations.

The basic Galerkin approximation is obtained by approximating the exact solution of the filtering SPDE (5) with a linear combination of basis functions $\phi_i(x)$, namely

$$\tilde{p}_t(x) := \sum_{i=1}^{\ell} c_i(t)\phi_i(x). \tag{12}$$

Ideally, the ϕ_i can be extended to indices $\ell + 1, \ell + 2, \dots, +\infty$ so as to form a basis of L^2 . The method can be sketched intuitively as follows. We could write the filtering Eq. (5) as

$$\left\langle -d p_t + \mathcal{L}_t^* p_t dt - \gamma_t^0(p_t) dt + \sum_{k=1}^d \gamma_t^k(p_t) \circ dY_t^k, \xi \right\rangle = 0$$

for all smooth L^2 test functions ξ such that the inner product exists. We replace this equation with the equation

$$\left\langle -d \tilde{p}_t + \mathcal{L}_t^* \tilde{p}_t dt - \gamma_t^0(\tilde{p}_t) dt + \sum_{k=1}^d \gamma_t^k(\tilde{p}_t) \circ dY_t^k, \phi_j \right\rangle = 0, \tag{13}$$

$j = 1, \dots, \ell.$

By substituting Eq. (12) in this last equation, using the linearity of the inner product in each argument and using integration by parts we obtain easily an equation for the combinatorics c , namely

$$\begin{aligned} \sum_{i=1}^{\ell} \langle \phi_i, \phi_j \rangle dc_i &= \sum_{i=1}^{\ell} \langle \phi_i, \mathcal{L} \phi_j \rangle c_i dt - \left\langle \gamma_t^0 \left(\sum_{h=1}^{\ell} c_h \phi_h \right), \phi_j \right\rangle \\ &+ \sum_{k=1}^d \left\langle \gamma_t^k \left(\sum_{h=1}^{\ell} c_h \phi_h \right), \phi_j \right\rangle \circ dY_t^k \end{aligned} \tag{13}$$

Consider now the projection filter with the following choice of the manifold. We use the convex hull of a set of basic L^2 probability densities q_1, \dots, q_{m+1} , namely the basic mixture family

$$\{p(\theta) := \theta_1 q_1 + \dots + \theta_m q_m + [1 - (\theta_1 + \dots + \theta_m)] q_{m+1}, \sum_{i=1}^m \theta_i < 1, \theta_i \geq 0 \text{ for all } i\}. \tag{14}$$

We can see easily that tangent vectors for the d_D structure and the matrix h are

$$v_j = q_j - q_{m+1}, \quad h_{i,j} = \langle q_i - q_{m+1}, q_j - q_{m+1} \rangle, \quad i, j = 1, \dots, m.$$

so that the metric is constant in θ . If we apply the d_D projection filter Eq. (9) with this choice of manifold, we see immediately by inspection that this equation coincides with the Galerkin method Eq. (13) if we take

$$\begin{aligned} \ell &= m + 1, \quad c_i = \theta_i \text{ and } \phi_i = q_i - q_{m+1} \text{ for } i = 1, \dots, m, \text{ and } c_{m+1} \\ &= 1, \quad \phi_{m+1} = q_{m+1}. \end{aligned}$$

The choice of the simple mixture is related to a choice of the L^2 basis in the Galerkin method. A typical choice could be based on Gaussian radial basis functions, see for example [34]. We have thus proven the following first main theoretical result of this paper:

Theorem 5.1 *For simple mixture families (14), the d_D projection filter (9) coincides with a Galerkin method (13) where the basis functions are the mixture components q .*

However, this equivalence holds only for the case where the manifold on which we project is the simple mixture family (14). More complex families, such as the ones we will use in the following, will not allow for a Galerkin-based filter and only the L^2 projection filter can be defined there. Note also that even in the simple case (14) our L^2 Galerkin/projection filter will be different from the Galerkin projection filter seen for example in [9], because we use Stratonovich calculus to project the Kushner–Stratonovich equation in L^2 metric. In [9], the Ito version of the Kushner–Stratonovich equation is used instead for the Galerkin method, but since Ito calculus does not work on manifolds, due to the second-order term moving the dynamics out of the tangent space, we use the Stratonovich version instead. The Ito-based and Stratonovich-based Galerkin projection filters will therefore differ for simple mixture families, and again, only the second one can be defined for manifolds of densities beyond the simplest mixture family.

6 Numerical software design

Equations (9) and (10) both give finite-dimensional stochastic differential equations that we hope will approximate well the solution to the full Kushner–Stratonovich equation. We wish to solve these finite-dimensional equations numerically and thereby obtain a numerical approximation to the nonlinear filtering problem. Because we are solving a low-dimensional system of equations we hope to end up with a more efficient scheme than a brute force finite difference approach. A finite difference approach can also be seen as a reduction of the problem to a finite-dimensional system. However, in a finite difference approach the finite-dimensional system still has a very large dimension, determined by the number of grid points into which one divides \mathbb{R}^n . By contrast the finite-dimensional manifolds we shall consider will be defined by only a handful of parameters. The specific solution algorithm will depend upon numerous choices: whether to use L^2 or Hellinger projection; which family of probability distributions to choose; how to parameterize that family; the representation of the functions f , σ and b ; how to perform the integrations which arise from the calculation of expectations and inner products; the numerical method selected to solve the finite-dimensional equations. To test the effectiveness of the projection idea, we have implemented a C++ engine which performs the numerical solution of the finite-dimensional equations and allows one to make various selections from the options above. Currently, our implementation is restricted to the case of the direct L^2 projection for a 1-dimensional

Table 1 UML for the FunctionRing interface

FunctionRing
+ add(f_1 : Function, f_2 : Function) : Function
+ multiply(f_1 : Function, f_2 : Function) : Function
+ multiply(s : Real, f : Function) : Function
+ differentiate(f : Function) : Function
+ integrate(f : Function) : Real
+ evaluate(f : Function) : Real
+ constantFunction(s : Real) : Function

Table 2 UML for the Manifold interface

Manifold
+ getRing() : FunctionRing
+ getDensity(θ) : Function
+ computeTangentVectors(θ : Point) : Function*
+ updatePoint(θ : Point, $\Delta\theta$: Real*) : Point
+ finalizePoint(θ : Point) : Point

state X and 1-dimensional noise W . However, the engine does allow one to experiment with various manifolds, parameterizations and functions f , σ and b . We use object-oriented programming techniques to allow this flexibility. Our implementation contains two key classes: **FunctionRing** and **Manifold**. To perform the computation, one must choose a data structure to represent elements of the function space. However, the most effective choice of representation depends upon the family of probability distributions one is considering and the functions f , σ and b . Thus, the C++ engine does not manipulate the data structure directly but instead works with the functions via the **FunctionRing** interface. An UML (Unified Modelling Language, <http://www.omg.org/spec/UML/>) outline of the **FunctionRing** interface is given in Table 1.

The other key abstraction is the **Manifold**. We give an UML representation of this abstraction in Table 2. For readers unfamiliar with UML, we remark that the $*$ symbol can be read “list”. For example, the `computeTangentVectors` function returns a list of functions. The **Manifold** uses some convenient internal representation for a point, the most obvious representation being simply the m -tuple $(\theta_1, \theta_2, \dots, \theta_m)$. On request the **Manifold** is able to provide the density associated with any point represented as an element of the **FunctionRing**. In addition, the **Manifold** can compute the tangent vectors at any point. The `computeTangentVectors` method returns a list of elements of the **FunctionRing** corresponding to each of the vectors $v_i = \frac{\partial p}{\partial \theta_i}$ in turn. If the point is represented as a tuple $\theta = (\theta_1, \theta_2, \dots, \theta_n)$, the method `updatePoint` simply adds the components of the tuple $\Delta\theta$ to each of the components of θ . If a different internal representation is used for the point, the method should make the equivalent change to this internal representation. The `finalizePoint` method is called by our algorithm at the end of every time step. At this point, the **Manifold** implementation can choose to change its parameterization for the state. Thus, the `finalizePoint` allows us (in principle at least) to use a more sophisticated atlas for the manifold than just a single

chart. One should not draw too close a parallel between these computing abstractions and similarly named mathematical abstractions. For example, the space of objects that can be represented by a given `FunctionRing` does not need to form a differential ring despite the `differentiate` method. This is because the `differentiate` function will not be called infinitely often by the algorithm below, so the functions in the ring do not need to be infinitely differentiable. Similarly the `finalizePoint` method allows the `Manifold` implementation more flexibility than simply changing chart. From one time step to the next, it could decide to use a completely different family of distributions. The interface even allows the dimension to change from one time step to the next. We do not currently take advantage of this possibility, but adaptively choosing the family of distributions would be an interesting topic for further research.

We now move to outlining our algorithm. The C++ engine is initialized with a `Manifold` object, a copy of the initial `Point` and `Function` objects representing f , σ and b . At each time point, the engine asks the manifold to compute the tangent vectors given the current point. Using the `multiply` and `integrate` functions of the class `FunctionRing`, the engine can compute the inner products of any two functions; hence, it can compute the metric matrix h_{ij} . Similarly, the engine can ask the manifold for the density function given the current point and can then compute \mathcal{L}^*p . Proceeding in this way, all the coefficients of dt and $\circ dY$ in Eq. (9) can be computed at any given point in time. Were Eq. (9) an Itô SDE one could now numerically estimate $\Delta\theta$, the change in θ over a given time interval Δ in terms of Δ and ΔY , the change in Y . One would then use the `updateState` method to compute the new point and then one could repeat the calculation for the next time interval. In other words, were Eq. (9) an Itô SDE we could numerically solve the SDE using the Euler scheme. However, Eq. (9) is a Stratonovich SDE so the Euler scheme is no longer valid. Various numerical schemes for solving stochastic differential equations are considered in [20] and [33]. One of the simplest is the Stratonovich–Heun method described in [20]. Suppose that one wishes to solve the SDE:

$$dy_t = f(y_t)dt + g(y_t) \circ dW_t$$

The Stratonovich–Heun method generates an estimate for the solution y_n at the n th time interval using the formulae:

$$\begin{aligned} Y_{n+1} &= y_n + f(y_n)\Delta + g(y_n)\Delta W_n \\ y_{n+1} &= y_n + \frac{1}{2}(f(y_n) + f(Y_{n+1}))\Delta + \frac{1}{2}(g(y_n) + g(Y_{n+1}))\Delta W_n \end{aligned}$$

In these formulae Δ is the size of the time interval and ΔW_n is the change in W . One can think of Y_{n+1} as being a prediction and the value y_{n+1} as being a correction. Thus, this scheme is a direct translation of the standard Euler–Heun scheme for ordinary differential equations. We can use the Stratonovich–Heun method to numerically solve Eq. (9). Given the current value θ_n for the state, compute an estimate for $\Delta\theta_n$ by replacing dt with Δ and dW with ΔW in Eq. (9). Using the `updateState` method compute a prediction Θ_{n+1} . Now compute a second estimate for $\Delta\theta_n$ using Eq. (9) in the state Θ_{n+1} . Pass the average of the two estimates to the `updateState` function to

obtain the the new state θ_{n+1} . At the end of each time step, the method `finalizeState` is called. This provides the manifold implementation the opportunity to perform checks such as validation of the state, to correct the normalization and, if desired, to change the representation it uses for the state. One small observation worth making is that the Eq. (9) contains the term h^{ij} , the inverse of the matrix h_{ij} . However, it is not necessary to actually calculate the matrix inverse in full. It is better numerically to multiply both sides of Eq. (9) by the matrix h_{ij} and then compute $d\theta$ by solving the resulting linear equations directly. This is the approach taken by our algorithm. As we have already observed, there is a wealth of choices one could make for the numerical scheme used to solve Eq. (9), we have simply selected the most convenient. The existing `Manifold` and `FunctionRing` implementations could be used directly by many of these schemes—in particular those based on Runge–Kutta schemes. In principle, one might also consider schemes that require explicit formulae for higher derivatives such as $\frac{\partial^2 p}{\partial \theta_i \partial \theta_j}$. In this case, one would need to extend the manifold abstraction to provide this information. Similarly one could use the same concepts to solve Eq. (10) where one uses the Hellinger projection. In this case, the `FunctionRing` would need to be extended to allow division. This would in turn complicate the implementation of the integrate function, which is why we have not yet implemented this approach.

7 The case of normal mixture families

We now apply the above framework to normal mixture families. Let \mathcal{R} denote the space of functions which can be written as finite linear combinations of terms of the form:

$$\pm x^n e^{ax^2+bx+c}$$

where n is non-negative integer and a, b and c are constants. \mathcal{R} is closed under addition, multiplication and differentiation, so it forms a differential ring. We have written an implementation of `FunctionRing` corresponding to \mathcal{R} . Although the implementation is mostly straightforward some points are worth noting. First, we store elements of our ring in memory as a collection of tuples (\pm, a, b, c, n) . Although one can write:

$$\pm x^n e^{ax^2+bx+c} = qx^n e^{ax^2+bx}$$

for appropriate q , the use of such a term in computer memory should be avoided as it will rapidly lead to significant rounding errors. A small amount of care is required throughout the implementation to avoid such rounding errors. Second, let us consider explicitly how to implement integration for this ring. Let us define u_n to be the integral of $x^n e^{-x^2}$. Using integration by parts one has:

$$u_n := \int_{-\infty}^{\infty} x^n e^{-x^2} dx = \frac{n-1}{2} \int_{-\infty}^{\infty} x^{n-2} e^{-x^2} dx = \frac{n-1}{2} u_{n-2}$$

Since $u_0 = \sqrt{\pi}$ and $u_1 = 0$ we can compute u_n recursively. Hence, we can analytically compute the integral of $p(x)e^{-x^2}$ for any polynomial p . By substitution, we can now integrate $p(x - \mu)e^{-(x-\mu)^2}$ for any μ . By completing the square we can analytically compute the integral of $p(x)e^{ax^2+bx+c}$ so long as $a < 0$. Putting all this together one has an algorithm for analytically integrating the elements of \mathcal{R} . Let \mathcal{N}^i denote the space of probability distributions that can be written as $\sum_{k=1}^i c_k e^{a_k x^2 + b_k x}$ for some real numbers a_k, b_k and c_k with $a_k < 0$. Given a smooth curve $\gamma(t)$ in \mathcal{N}^i we can write:

$$\gamma(t) = \sum_{k=1}^i c_k(t) e^{a_k(t)x^2 + b_k(t)x}.$$

We can then compute:

$$\frac{d\gamma}{dt} = \sum_{k=1}^i \left(\left(\frac{da_k}{dt} x^2 + \frac{db_k}{dt} x \right) c_k e^{a_k x^2 + b_k x} + \frac{dc_k}{dt} e^{a_k x^2 + b_k x} \right) \in \mathcal{R}$$

We deduce that the tangent vectors of any smooth submanifold of \mathcal{N}^i must also lie in \mathcal{R} . In particular this means that our implementation of `FunctionRing` will be sufficient to represent the tangent vectors of any manifold consisting of finite normal mixtures. Combining these ideas, we obtain the second main theoretical result of the paper.

Theorem 7.1 *Let θ be a parameterization for a family of probability distributions all of which can be written as a mixture of at most i Gaussians. Let $f, a = \sigma^2$ and b be functions in the ring \mathcal{R} . In this case, one can carry out the direct L^2 projection algorithm for the problem given by Eq. (1) using analytic formulae for all the required integrations.*

Although the condition that f, a and b lie in \mathcal{R} may seem somewhat restrictive, when this condition is not met one could use Taylor expansions to find approximate solutions, although in such case rigorous convergence results need to be established. Although the choice of parameterization does not affect the choice of `FunctionRing`, it does affect the numerical behaviour of the algorithm. In particular if one chooses a parameterization with domain a proper subset of \mathbb{R}^m , the algorithm will break down the moment the point θ leaves the domain. With this in mind, in the numerical examples given later in this paper we parameterize normal mixtures of k Gaussians with a parameterization defined on the whole of \mathbb{R}^n . We describe this parameterization below.

Label the parameters ξ_i (with $1 \leq i \leq k - 1$), x_1, y_i (with $2 \leq i \leq k$) and s_i (with $1 \leq i \leq k$). This gives a total of $3k - 1$ parameters. So we can write

$$\theta = (\xi_1, \dots, \xi_{k-1}, x_1, y_2, \dots, y_k, s_1, \dots, s_k)$$

Given a point θ define variables as follows:

$$\begin{aligned} \lambda_1 &= \text{logit}^{-1}(\xi_1) \\ \lambda_i &= \text{logit}^{-1}(\xi_i)(1 - \lambda_1 - \lambda_2 - \dots - \lambda_{i-1}) \quad (2 \leq i \leq k - 1) \end{aligned}$$

$$\begin{aligned} \lambda_k &= 1 - \lambda_1 - \lambda_2 - \dots - \lambda_{k-1} \\ x_i &= x_{i-1} + e^{y_i} \quad (2 \leq i \leq k) \\ \sigma_i &= e^{s_i} \end{aligned}$$

where the logit function sends a probability $p \in [0, 1]$ to its log odds, $\ln(p/1 - p)$. We can now write the density associated with θ as:

$$p(x) = \sum_{i=1}^k \lambda_i \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(x - x_i)^2}{2\sigma_i^2}\right)$$

We do not claim this is the best possible choice of parameterization, but it certainly performs better than some more naïve parameterizations with bounded domains of definition. We will call the direct L^2 projection algorithm onto the normal mixture family given with this projection the L2NM projection filter.

7.1 Comparison with the Hellinger exponential (HE) projection algorithm

A similar algorithm is described in [17, 18] for projection using the Hellinger metric onto an exponential family. We refer to this as the HE projection filter. It is worth highlighting the key differences between our algorithm and the exponential projection algorithm described in [17].

- In [17], only the special case of the cubic sensor was considered. It was clear that one could in principle adapt the algorithm to cope with other problems, but there remained symbolic manipulation that would have to be performed by hand. Our algorithm automates this process using the `FunctionRing` abstraction.
- When one projects onto an exponential family, the stochastic term in Eq. (10) simplifies to a term with constant coefficients. This means it can be viewed equally well as either an Itô or Stratonovich SDE. The practical consequence of this is that the HE algorithm can use the Euler–Maruyama scheme rather than the Stratonovich–Heun scheme to solve the resulting stochastic ODE’s. Moreover, in this case, the Euler–Maruyama scheme coincides with the generally more precise Milstein scheme.
- In the case of the cubic sensor, the HE algorithm requires one to numerically evaluate integrals such as:

$$\int_{-\infty}^{\infty} x^n \exp(\theta_1 + \theta_2 x + \theta_3 x^2 + \theta_4 x^4) dx$$

where the θ_i are real numbers. Performing such integrals numerically considerably slows the algorithm. In effect one ends up using a rather fine discretization scheme to evaluate the integral and this somewhat offsets the hoped for advantage over a finite difference method.

8 Numerical results

In this section, we compare the results of using the direct L^2 projection filter onto a mixture of normal distributions with other numerical methods. In particular we compare it with:

1. A finite difference method using a fine grid which we term the exact filter. Various convergence results are known ([35] and [36]) for this method. In the simulations shown below we use a grid with 1000 points on the x -axis and 5000 time points. In our simulations, we could not visually distinguish the resulting graphs when the grid was refined further justifying us in considering this to be extremely close to the exact result. The precise algorithm used is as described in the section on “Partial Differential Equations Methods” in chapter 8 of Bain and Crisan [12].
2. The extended Kalman filter (EK) This is a somewhat heuristic approach to solving the nonlinear filtering problem but which works well so long as one assumes the system is almost linear. It is implemented essentially by linearising all the functions in the problem and then using the exact Kalman filter to solve this linear problem—the details are given in [12]. The EK filter is widely used in applications and so provides a standard benchmark. However, it is well known that it can give wildly inaccurate results for nonlinear problems so it should be unsurprising to see that it performs badly for most of the examples we consider.
3. The HE projection filter. In fact, we have implemented a generalization of the algorithm given in [19] that can cope with filtering problems where b is an arbitrary polynomial, σ is constant and $f = 0$. Thus, we have been able to examine the performance of the exponential projection filter over a slightly wider range of problems than have previously been considered.

To compare these methods, we have simulated solutions of the Eq. (1) for various choices of f , σ and b . We have also selected a prior probability distribution p_0 for X and then compared the numerical estimates for the probability distribution p at subsequent times given by the different algorithms. In the examples below we have selected a fixed value for the initial state X_0 rather than drawing at random from the prior distribution. This should have no more impact upon the results than does the choice of seed for the random number generator. Since each of the approximate methods can only represent certain distributions accurately, we have had to use different prior distributions for each algorithm. To compare the two projection filters, we have started with a polynomial exponential distribution for the prior and then found a nearby mixture of normal distributions. This nearby distribution was found using a gradient search algorithm to minimize the numerically estimated L^2 norm of the difference of the normal and polynomial exponential distributions. As indicated earlier, polynomial exponential distributions and normal mixtures are qualitatively similar so the prior distributions we use are close for each algorithm. For the extended Kalman filter, one has to approximate the prior distribution with a single Gaussian. We have done this by moment matching. Inevitably this does not always produce satisfactory results. For the exact filter, we have used the same prior as for the L^2 projection filter.

8.1 The linear filter

The first test case we have examined is the linear filtering problem. In this case, the probability density will be a Gaussian at all times—hence if we project onto the two-dimensional family consisting of all Gaussian distributions there should be no loss of information. Thus, both projection filters should give exact answers for linear problems. This is indeed the case and gives some confidence in the correctness of the computer implementations of the various algorithms.

8.2 The quadratic sensor

The second test case we have examined is the quadratic sensor. This is problem (1) with $f = 0$, $\sigma = c_1$ and $b(x) = c_2x^2$ for some positive constants c_1 and c_2 . In this problem, the non-injectivity of b tends to cause the distribution at any time to be bimodal. To see why, observe that the sensor provides no information about the sign of x , once the state of the system has passed through 0 we expect the probability density to become approximately symmetrical about the origin. Since we expect the probability density to be bimodal for the quadratic sensor, it makes sense to approximate the distribution with a linear combination of two Gaussian distributions. In Fig. 2 we show the probability density as computed by three of the algorithms at 10 different time points for a typical quadratic sensor problem. To reduce clutter, we have not plotted the results for the exponential filter. The prior exponential distribution used for this simulation was $p(x) = \exp(0.25 - x^2 + x^3 - 0.25x^4)$. The initial state was $X_0 = 0$ and $Y_0 = 0$. As one can see the probability densities computed using the exact filter and the L2NM filter become visually indistinguishable when the state moves away from the origin. The extended Kalman filter is, as one would expect, completely unable to cope with these bimodal distributions. In this case, the extended Kalman filter is simply representing the larger of the two modes. In Fig. 5, we have plotted the L^2 residuals for the different algorithms when applied to the quadratic sensor problem. We define the L^2 residual to be the L^2 norm of the difference between the exact filter distribution and the estimated distribution.

$$L^2 \text{ residual} = \left(\int |p_{\text{exact}} - p_{\text{approx}}|^2 d\mu \right)^{\frac{1}{2}}$$

As can be seen, the L2NM projection filter outperforms the HE projection filter when applied to the quadratic sensor problem. Notice that the L^2 residuals are initially small for both the HE and the L2NM filter. The superior performance of the L2NM projection filter in this case stems from the fact that one can more accurately represent the distributions that occur using the normal mixture family than using the polynomial exponential family. If preferred one could define a similar notion of residual using the Hellinger metric. The results would be qualitatively similar. One interesting feature of Fig. 5 is that the error remains bounded in size when one might expect the error to accumulate over time. This suggests that the arrival of new measurements is gradually correcting for the errors introduced by the approximation.

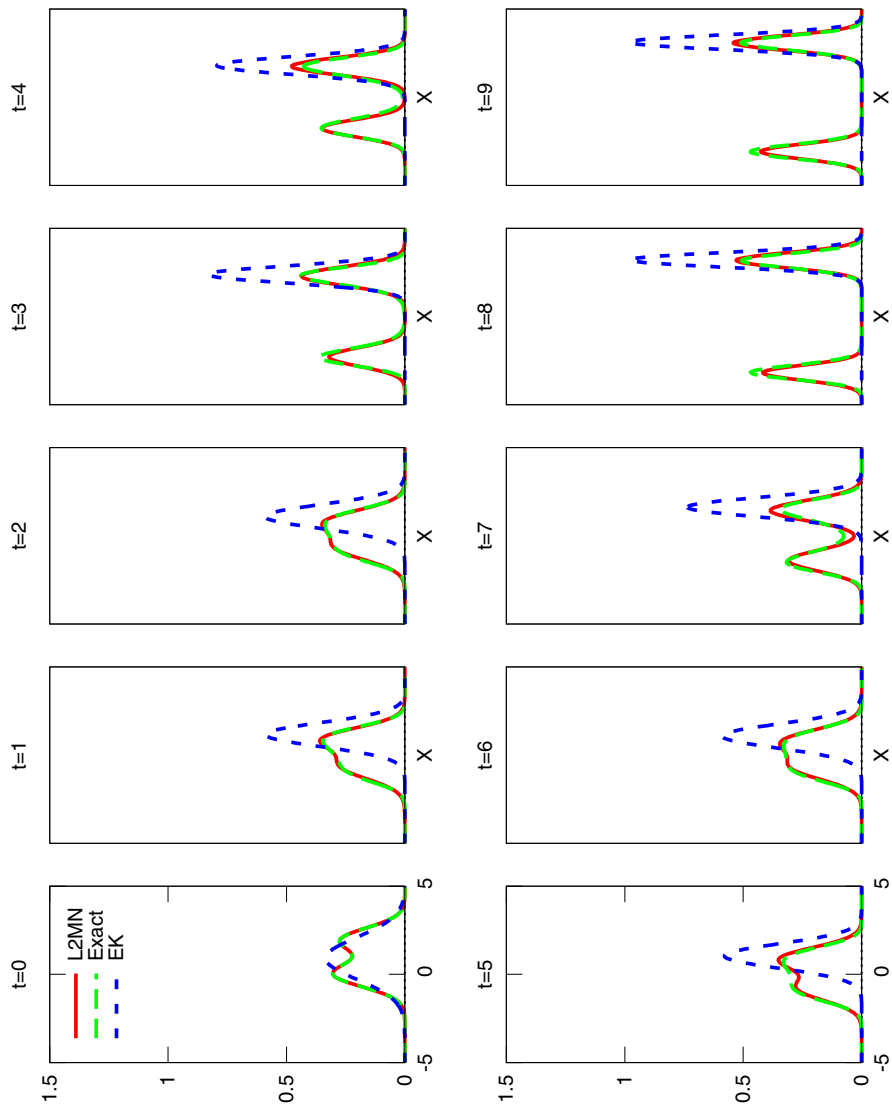


Fig. 2 Estimated probability densities at 10 time points for the problem $b(x) = x^2$

8.3 The cubic sensor

A third test case we have considered is the general cubic sensor. In this problem, one has $f = 0$, $\sigma = c_1$ for some constant c_1 and b is some cubic function. The case when b is a multiple of x^3 is called the cubic sensor and was used as the test case for the exponential projection filter using the Hellinger metric considered in [19]. It is of interest because it is the simplest case where b is injective but where it is known that the problem cannot be reduced to a finite-dimensional stochastic differ-

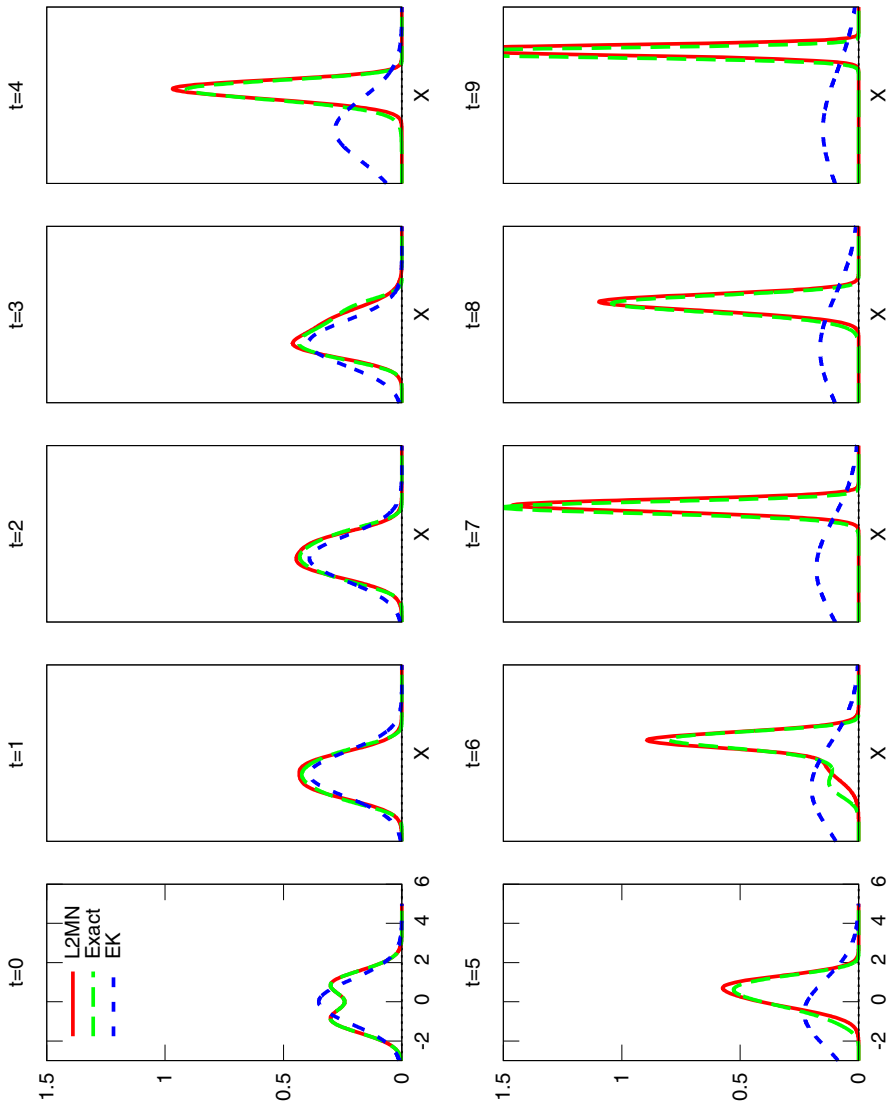


Fig. 3 Estimated probability densities at 10 time points for the problem $b(x) = x^3 - x$

ential equation [26,42]. It is known from earlier work that the exponential filter gives excellent numerical results for the cubic sensor. Our new implementations allow us to examine the general cubic sensor. In Fig. 3, we have plotted example probability densities over time for the problem with $f = 0, \sigma = 1$ and $b = x^3 - x$. With two turning points for b , this problem is very far from linear. As can be seen in Fig. 3, the L2NM projection remains close to the exact distribution throughout. A mixture of only two Gaussians is enough to approximate quite a variety of differently shaped distributions with perhaps surprising accuracy. As expected, the extended Kalman

filter gives poor results until the state moves to a region where b is injective. The results of the exponential filter have not been plotted in Fig. 3 to reduce clutter. It gave similar results to the L2NM filter. The prior polynomial exponential distribution used for this simulation was $p(x) = \exp(0.5x^2 - 0.25x^4)$. The initial state was $X_0 = 0$, which is one of the modes of prior distribution. The initial value for Y_0 was taken to be 0. One new phenomenon that occurs when considering the cubic sensor is that the algorithm sometimes abruptly fails. This is true for both the L2NM projection filter and the HE projection filter. To show the behaviour over time more clearly, in Fig. 4 we have shown a plot of the mean and standard deviation as estimated by the L2NM projection filter against the actual mean and standard deviation. We have also indicated the true state of the system. The mean for the L2MN filter drops to 0 at approximately time 7. It is at this point that the algorithm has failed. What has happened is that as the state has moved to a region where the sensor is reasonably close to being linear, the probability distribution has tended to a single normal distribution. Such a distribution lies on the boundary of the family consisting of a mixture of two normal distributions. As we approach the boundary, h_{ij} ceases to be invertible causing the failure of the algorithm. Analogous phenomena occur for the exponential filter. The result of running numerous simulations suggests that the HE filter is rather less robust than the L2NM projection filter. The typical behaviour is that the exponential filter maintains a very low residual right up until the point of failure. The L2NM projection filter on the other hand tends to give slightly inaccurate results shortly before failure and can often correct itself without failing. This behaviour can be seen in Fig. 4. In this figure, the residual for the exponential projection remains extremely low until the algorithm fails abruptly—this is indicated by the vertical dashed line. The L2NM filter on the other hand deteriorates from time 6 but only fails at time 7. The L^2 residuals of the L2MN method are rather large between times 6 and 7, but note that the accuracy of the estimates for the mean and standard deviation in Fig. 4 remains reasonable throughout this time. To understand this note that for two normal distributions with means a distance x apart, the L^2 distance between the distributions increases as the standard deviations of the distributions drop. Thus, the increase in L^2 residuals between times 6 and 7 is to a large extent due to the drop in standard deviation between these times. As a result, one may feel that the L^2 residual does not capture precisely what it means for an approximation to be “good”. In the next section, we will show how to measure residuals in a way that corresponds more closely to the intuitive idea of them having visually similar distribution functions. In practice, one’s definition of a good approximation will depend upon the application. Although one might argue that the filter is in fact behaving reasonably well between times 6 and 7 it does ultimately fail. There is an obvious fix for failures like this. When the current point is sufficiently close to the boundary of the manifold, as in the above example of the mixture of two normals trying to collapse to a single normal, simply approximate the distribution with an element of the boundary. In other words, approximate the distribution using a mixture of fewer Gaussians. Since this means moving to a lower dimensional family of distributions, the numerical implementation will be more efficient on the boundary. This will provide a temporary fix the failure of the algorithm, but it raises another problem: as the state moves back into a region where the problem is highly nonlinear, how can one decide how to leave the boundary and start adding addi-

tional Gaussians back into the mixture? We hope to address this question in a future paper.

9 Comparison with particle methods

Particle methods approximate the probability density p using discrete measures of the form:

$$\sum_i a_i(t) \delta_{v_i(t)}$$

These measures are generated using a Monte Carlo method. The measure can be thought of as the empirical distributions associated with randomly located particles at position $v_i(t)$ and of stochastic mass $a_i(t)$. Particle methods are currently some of the most effective numerical methods for solving the filtering problem. See [12] and the references therein for details of specific particle methods and convergence results. The first issue in comparing projection methods with particle methods is that, as a linear combination of Dirac masses, one can only expect a particle method to converge weakly to the exact solution. In particular, the L^2 metric and the Hellinger metric are both inappropriate measures of the residual between the exact solution and a particle approximation. Indeed, the L^2 distance is not defined and the Hellinger distance will always take the value $\sqrt{2}$. To combat this issue, we will measure residuals using the Lévy metric. If p and q are two probability measures on \mathbb{R} and P and Q are the associated cumulative distribution functions, then the Lévy metric is defined by:

$$d_L(p, q) = \inf\{\epsilon : P(x - \epsilon) - \epsilon \leq Q(x) \leq P(x + \epsilon) + \epsilon \quad \forall x\}$$

This can be interpreted geometrically as the size of the largest square with sides parallel to the coordinate axes that can be inserted between the completed graphs of the cumulative distribution functions (the completed graph of the distribution function is simply the graph of the distribution function with vertical line segments added at discontinuities). The Lévy metric can be seen as a special case of the Lévy–Prokhorov metric. This can be used to measure the distance between measures on a general metric space. For Polish spaces, the Lévy–Prokhorov metric metrises the weak convergence of probability measures [11]. Thus, the Lévy metric provides a reasonable measure of the residual of a particle approximation. We will call residuals measured in this way Lévy residuals. A second issue in comparing projection methods with particle methods is deciding how many particles to use for the comparison. A natural choice is to compare a projection method onto an m -dimensional manifold with a particle method that approximates the distribution using $\lceil (m + 1)/2 \rceil$ particles. In other words, equate the dimension of the families of distributions used for the approximation. A third issue is deciding which particle method to choose for the comparison from the many algorithms that can be found in the literature. We can work around this issue by calculating the best possible approximation to the exact distribution that can be made using $\lceil (m + 1)/2 \rceil$ Dirac masses. This approach will substantially underestimate the

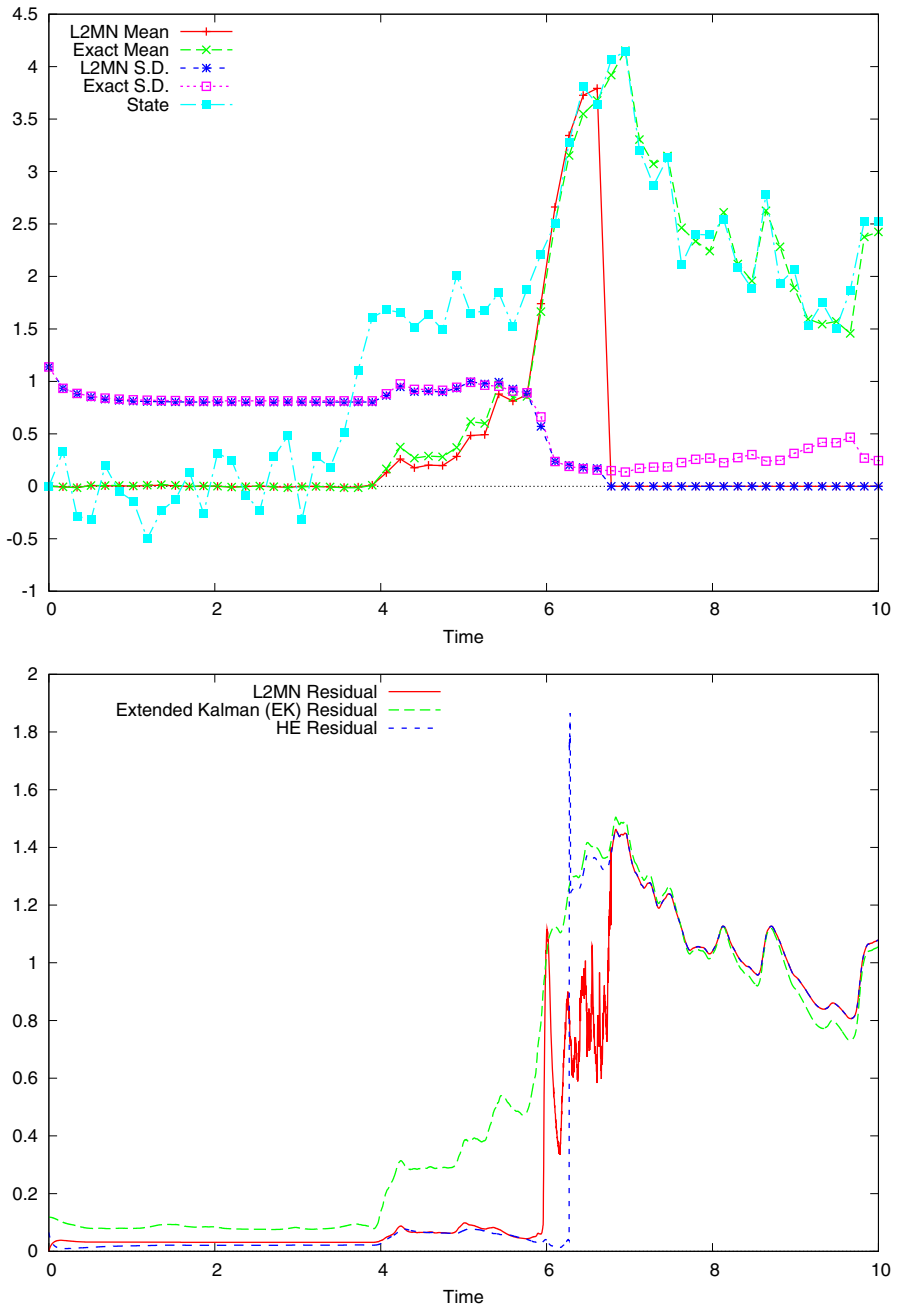


Fig. 4 Estimates for the mean and standard deviation (*above*) and L^2 residuals (*below*) for the problem $b(x) = x^3 - x$

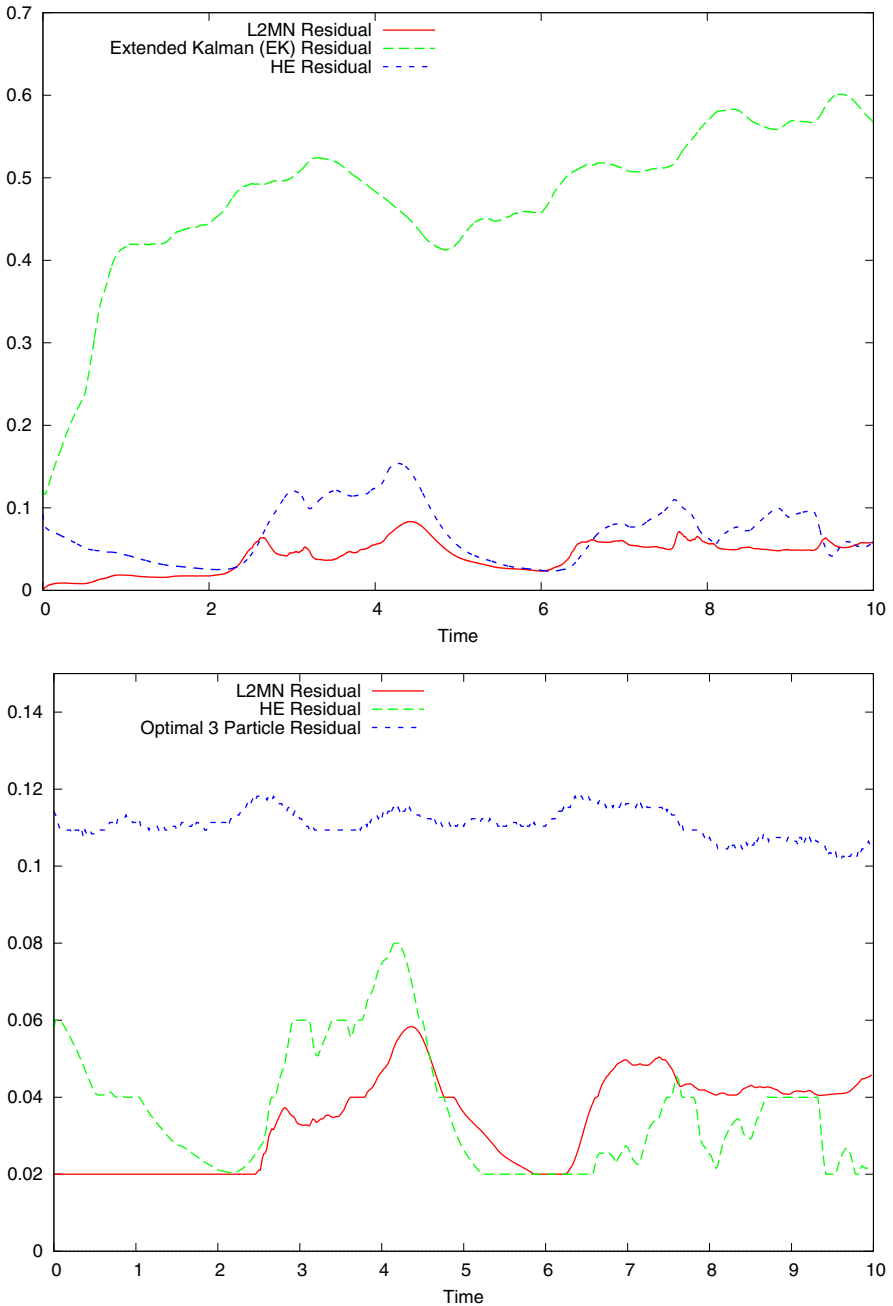


Fig. 5 L^2 (above) and Lévy (below) residuals for the problem $b(x) = x^2$

Lévy residual of a particle method: being Monte Carlo methods, large numbers of particles would be required in practice. In Fig. 5, we have plotted bounds on the Lévy residuals for the two projection methods for the quadratic sensor. Since mixtures of two normal distributions lie in a 5-dimensional family, we have compared these residuals with the best possible Lévy residual for a mixture of three Dirac masses. To compute the Lévy residual between two functions, we have approximated first approximated the cumulative distribution functions using step functions. We have used the same grid for these steps as we used to compute our “exact” filter. We have then used a brute force approach to compute a bound on size of the largest square that can be placed between these step functions. Thus, if we have used a grid with n points to discretize the x -axis, we will need to make n^2 comparisons to estimate the Lévy residual. More efficient algorithms are possible, but this approach is sufficient for our purposes. The maximum accuracy of the computation of the Lévy metric is constrained by the grid size used for our “exact” filter. Since the grid size in the x direction for our “exact” filter is 0.01, our estimates for the projection residuals are bounded below by 0.02. The computation of the minimum residual for a particle filter is a little more complex. Let $\text{minEpsilon}(F, n)$ denote the minimum Lévy distance between a distribution with cumulative distribution F and a distribution of n particles. Let $\text{minN}(F, \epsilon)$ denote the minimum number of particles required to approximate F with a residual of less than ϵ . If we can compute minN we can use a line search to compute minEpsilon . To compute $\text{minN}(F, \epsilon)$ for an increasing step function F with $F(-\infty) = 0$ and $F(\infty) = 1$, one needs to find the minimum number of steps in a similar increasing step function G that is never further than ϵ away from F in the L^∞ metric. One constructs candidate step functions G by starting with $G(-\infty) = 0$ and then moving along the x -axis adding in additional steps as required to remain within a distance ϵ . An optimal G is found by adding in steps as late as possible and, when adding a new step, making it as high as possible. In this way, we can compute minN and minEpsilon for step functions F . We can then compute bounds on these values for a given distribution by approximating its cumulative density function with a step function. As can be seen, the exponential and mixture projection filters have similar accuracy as measured by the Lévy residual and it is impossible to match this accuracy using a model containing only 3 particles.

10 Comparison with robust Zakai implementation using Hermite functions

Luo and Yau [38–40] propose solving the filtering problem in real time by solving the robust Zakai equation using a spectral method based on Hermite functions. We have implemented such a filter. We found that, when one uses 45 Hermite basis functions as recommended in the papers, this approach produces excellent results which are essentially indistinguishable from our own “exact” filter implementation. This provides further evidence for the effectiveness of Luo and Yau’s approach.

For one-dimensional problems, this Hermite-based spectral method is more than sufficient to solve the filtering problem in real time. However, if we apply exactly the same approach to a filtering problem with n -dimensional state space one might estimate

that we would need 45^n basis functions. This is the familiar curse of dimensionality. Therefore, it is interesting to know how the Hermite spectral method degrades as one reduces the number of basis functions. One might expect that the approach would deteriorate gradually as the number of basis functions is decreased, but in fact our numerical results indicate that it fails rapidly once the number of basis functions used is dropped to about 25. We have not plotted the results since they are more easily summarized verbally: with 45 basis functions the Hermite spectral method is excellent (as shown also by Luo and Yau above), whereas with less than 20 basis functions it does not provide a useful approximation.

By contrast, the projection filter above is able to find reasonably accurate solutions using only a 5-dimensional manifold. We may summarize the different approaches as follows. Spectral methods such as Luo and Yau's provide an approach that gives a practical way of finding real-time solutions in very low dimensions. Particle filters provide a method of finding approximate solutions in large dimensions but are difficult to apply in real time, even for low-dimensional systems. Projection methods provide a promising avenue for finding approximate solutions to medium-dimensional filtering problems in real time.

11 Conclusions

Projection onto a family of normal mixtures using the L^2 metric (L2NM) allows one to approximate the solutions of the nonlinear filtering problem with surprising accuracy using only a small number of component distributions. In this regard, this filter behaves in a very similar fashion to the projection onto an exponential family using the Hellinger metric that has been considered previously. The L2NM projection filter has one important advantage over the Hellinger exponential (HE) projection filter: for problems with polynomial coefficients all required integrals can be calculated analytically. Problems with more general coefficients can be addressed using Taylor series. One expects this to translate into a better performing algorithm—particularly if the approach is extended to higher dimensional problems.

We tested both filters against the optimal filter in simple but interesting systems, and provided a metric to compare the performance of each filter with the optimal one.

We also tested both filters against a particle method, showing that with the same number of parameters the L2NM filter outperforms the best possible particle method in Levy metric.

We designed a software structure and populated it with models that make the L2NM filter quite appealing from a numerical and computational point of view.

Areas of future research that we hope to address include: the relationship between the projection approach and existing numerical approaches to the filtering problem; the convergence of the algorithm; improving the stability and performance of the algorithm by adaptively changing the parameterization of the manifold; numerical simulations in higher dimensions; more generally, we are investigating whether a new type of projection, building on the Ito stochastic calculus structure, could be suited to derive approximate equations.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Aggrawal J (1974) Sur l'information de Fisher. In: Kampé de Fériet J (ed) *Théories de l'Information*. Springer-Verlag, Berlin, pp 111–117
2. Ahmed NU (1998) *Linear and nonlinear filtering for scientists and engineers*. World Scientific, Singapore
3. Ahmed NU, Radaideh S (1997) A powerful numerical technique solving the Zakai equation for nonlinear filtering. *Dyn Control* 7(3):293–308
4. Alspach DL, Sorenson HW (1972) Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans Automat Control AC-17* (4):439–448
5. Amari S (1985) *Differential-geometrical methods in statistics*, Lecture notes in statistics. Springer-Verlag, Berlin
6. Armstrong J, Brigo D (2013) Stochastic filtering via L^2 projection on mixture manifolds with computer algorithms and numerical examples. Available at arXiv.org
7. Bagchi A, Karandikar RL (1994) White noise theory of robust nonlinear filtering with correlated state and observation noises. *Syst Control Lett* 23:137–148
8. Balakrishnan AV (1980) Nonlinear white noise theory. In: Krishnaiah PR (ed) *Multivariate Analysis—V*. North-Holland, Amsterdam, pp 97–109
9. Beard R, Gunther J (1997) Galerkin Approximations of the Kushner Equation in Nonlinear Estimation. Working Paper, Brigham Young University
10. Bensoussan A, Glowinski R, Rascanu A (1990) Approximation of the Zakai equation by the splitting up method. *SIAM J Control Optim* 28(6):1420–1431
11. Billingsley Patrick (1999) *Convergence of Probability Measures*. Wiley, New York
12. Bain A, Crisan D (2010) *Fundamentals of Stochastic Filtering*. Springer-Verlag, Heidelberg
13. Barndorff-Nielsen OE (1978) *Information and Exponential Families*. Wiley, New York
14. Beneš VE (1981) Exact finite-dimensional filters for certain diffusions with nonlinear drift. *Stochastics* 5:65–92
15. Beneš VE, Elliott RJ (1996) Finite-dimensional solutions of a modified Zakai equation. *Math Signals Syst* 9(4):341–351
16. Brigo D (2012) The direct L^2 geometric structure on a manifold of probability densities with applications to Filtering. Available at arXiv.org
17. Brigo D, Hanzon B, LeGland F (1998) A differential geometric approach to nonlinear filtering: the projection filter. *IEEE Trans Autom Control* 43:247–252
18. Brigo D, Hanzon B, Le Gland F (1999) Approximate nonlinear filtering by projection on exponential manifolds of densities. *Bernoulli* 5:495–534
19. Brigo D (1996) *Filtering by Projection on the Manifold of Exponential Densities*. PhD Thesis, Free University of Amsterdam
20. Burrage K, Burrage PM, Tian T (2004) Numerical methods for strong solutions of stochastic differential equations: an overview. *Proc R Soc Lond A* 460:373–402
21. Clark JMC (1978) The design of robust approximations to the stochastic differential equations of nonlinear filtering. In: Skwirzynski JK (ed) *Communication Systems and Random Process Theory*, NATO Advanced Study Institute Series. (Sijthoff and Noordhoff, Alphen aan den Rijn)
22. Davis MHA (1980) On a multiplicative functional transformation arising in nonlinear filtering theory. *Z Wahrsch Verw Geb* 54:125–139
23. Frey R, Schmidt T, Xu L (2013) On Galerkin approximations for the Zakai equation with diffusive and point process observations. *SIAM J Numer Anal* 51(4):2036–2062
24. Germani A, Picconi M (1984) A Galerkin approximation for the Zakai equation. In: Thoft-Christensen P (ed) *System Modelling and Optimization* (Copenhagen, 1983), *Lecture Notes in Control and Information Sciences*, Vol 59, Springer-Verlag, Berlin, pp 415–423
25. Hanzon B (1987) A differential-geometric approach to approximate nonlinear filtering. In: Dodson CTJ (ed) *Geometrization of Statistical Theory*, ULMD Publications, University of Lancaster, pp 219–223

26. Hazewinkel M, Marcus SI, Sussmann HJ (1983) Nonexistence of finite dimensional filters for conditional statistics of the cubic sensor problem. *Syst Control Lett* 3:331–340
27. Ito K (1996) Approximation of the Zakai equation for nonlinear filtering. *SIAM J Control Optim* 34(2):620–634
28. Jacod J, Shiryaev AN (1987) Limit theorems for stochastic processes. *Grundlehren der Mathematischen Wissenschaften*, vol 288. Springer-Verlag, Berlin
29. Jazwinski AH (1970) Stochastic processes and filtering theory. Academic Press, New York
30. Fujisaki M, Kallianpur G, Kunita H (1972) Stochastic differential equations for the non linear filtering problem. *Osaka J. Math.* 9(1):19–40
31. Kallianpur G, Karandikar RL (1983) A finitely additive white noise approach to nonlinear filtering. *Appl Math Optim* 10:159–186
32. Khasminskii RZ (1980) Stochastic stability of differential equations. Alphen aan den Rijn
33. Kloeden PE, Platen E (1999) Numerical solution of stochastic differential equations. Springer, Berlin
34. Kormann K, Larsson E (2014) A Galerkin radial basis function method for the Schroedinger Equation. *SIAM J Sci Comput* 35(6):A2832–A2855
35. Kushner HJ (1990) Weak convergence methods and singularly perturbed stochastic control and filtering problems, volume 3 of *Systems & Control: Foundations & Applications*. Birkhäuser Boston
36. Kushner HJ, Huang H (1986) Approximate and limit results for nonlinear filters with wide bandwidth observation noise. *Stochastics* 16(1–2):65–96
37. Liptser RS, Shiryaev AN (1978) Statistics of random processes I, general theory. Springer Verlag, Berlin
38. Luo X, Yau SS-T (2012) A novel algorithm to solve the robust DMZ equation in real time. In: *Proceedings of the 51th Conference in Decision and Control (CDC)*, pp 606–611
39. Luo X, Yau SS-T (2013) Complete real time solution of the general nonlinear filtering problem without memory. *IEEE Trans Automat Control* 58(10):2563–2578
40. Luo X, Yau SS-T (2013) Hermite spectral method to 1D forward Kolmogorov equation and its application to nonlinear filtering problems. *IEEE Trans Automat Control* 58(10):2495–2507
41. Nowak LD, Paslawska-Poludniak M, Twardowska K (2010) On the convergence of the Wavelet-Galerkin method for nonlinear filtering. *Int J Appl Math Comput Sci* 20(1):93–108
42. Ocone D (1988) Probability densities for conditional statistics in the cubic sensor problem. *Math Control Signals Syst* 1(2):183–202
43. Pardoux E (1979) Stochastic partial differential equations and filtering of diffusion processes. *Stochastics* 3:127–167
44. Pistone G, Sempi C (1995) An Infinite dimensional geometric structure on the space of all the probability measures equivalent to a given one. *Ann Stat* 23(5):1543–1561