FLUX: Progressive State Estimation Based on Zakai-type Distributed Ordinary Differential Equations

Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany

Abstract

We propose a homotopy continuation method called FLUX for approximating complicated probability density functions. It is based on progressive processing for smoothly morphing a given density into the desired one. Distributed ordinary differential equations (DODEs) with an artificial time $\gamma \in [0,1]$ are derived for describing the evolution from the initial density to the desired final density. For a finite-dimensional parametrization, the DODEs are converted to a system of ordinary differential equations (SODEs), which are solved for $\gamma \in [0,1]$ and return the desired result for $\gamma = 1$. This includes parametric representations such as Gaussians or Gaussian mixtures and nonparametric setups such as sample sets. In the latter case, we obtain a particle flow between the two densities along the artificial time.

FLUX is applied to state estimation in stochastic nonlinear dynamic systems by gradual inclusion of measurement information. The proposed approximation method (1) is fast, (2) can be applied to arbitrary nonlinear systems and is not limited to additive noise, (3) allows for target densities that are only known at certain points, (4) does not require optimization, (5) does not require the solution of partial differential equations, and (6) works with standard procedures for solving SODEs. This manuscript is limited to the one-dimensional case and a fixed number of parameters during the progression. Future extensions will include consideration of higher dimensions and on the fly adaption of the number of parameters.

1. Introduction

We consider the approximation of complicated probability density functions (pdfs) by densities with convenient finite-dimensional representations. This includes parametric density approximation such as Gaussian densities or Gaussian mixture densities and nonparametric representations such as sample sets. Three problem classes are investigated: (1) The fundamental problem of approximating a given probability density function that may only be given at certain points. (2) The problem of approximating posterior densities in estimating a hidden state based on prior densities and given measurements. (3) The problem of propagating a state estimate through a discrete-time stochastic nonlinear dynamic system.

Optimal approximation of complicated probability density functions is a hard problem and typically involves the minimization of a suitable distance measure between the original density and its approximation. The minimization problem is in general nonlinear and nonconvex, which precludes efficient solution procedures and calls for iterative optimization methods. Depending on the selected starting points, the run time varies and one might end up in local minima.

In this manuscript, our goal is to find a flow between a simple density with a given approximation and the target density. This flow is then used to move the parameters of the approximating density in such a way that they finally approximate the target density.

1.1. State of the Art

1.1.1. Overview of Relevant Literature

When using a particle representation for the posterior probability density functions in a Bayesian update step, it is well known that a narrow likelihood function corresponding to a precise measurement typically leads to particle degeneration [1]. This means that due to the downweighting with the likelihood function, only a few particles maintain nonzero weights so that all the other particles are wasted as they do not contribute to the density approximation.

Particle degeneration can be mitigated with a discrete set of so-called bridging densities that allow a gradual introduction of the measurement by employing them successively with intermediate resampling as opposed to the direct multiplication of a particle set with the original likelihood [2]. This strategy is used by annealed sampling methods [3, 4, 5]. They use a form of MCMC or sample movement to cope with the particle degeneracy. Particle-flow filters work in a fundamentally different fashion as they do not employ particle weighting at all.

The Daum-Huang filters [6, 7, 8] use a homotopy between the logarithms of the unnormalized prior and posterior densities, which is used to calculate a particle flow. Different versions have been proposed, see [9, 10, 11]. The flow moves particles towards regions of a high posterior probability. Its calculation entails the solution of a partial differential equation, which is computationally complex. Two types of Daum-Huang filters are compared with standard filter approaches in [12]. To cope with the high computational complexity of particle flows, an approximate Gaussian flow is introduced in [13]. Here, the particles are not used directly for representing the posterior density. Instead, they are used as an input to an importance sampler.

Related to particle-flow filters are the so-called feedback particle filters [14] that apply a feedback mechanism in order to move the particles. According to [15], these filters are a generalization of linear-regression filters such as the UKF [16] or the smart-sampling Kalman filter first proposed in [17], refined in [18], and optimized by enforcing symmetric samples in [19]. The feedback gains of the feedback particle filters are obtained by minimizing a cost function based on the Kullback-Leibler divergence. An evaluation of feedback particle filters is given in [20].

1.1.2. Own Prior Work

A novel estimation method based on progressively introducing measurement information in order to perform morphing between a given density and a desired density, in this case, the posterior resulting from a Bayesian update step, is introduced in [21, 22]. A generalization to multi-dimensional states is given in [23]. In [24], this method was generalized to different distance measures. Further applications of progressive processing methods besides state estimation are in moment calculation [25] and Gaussian mixture reduction [26].

A progressive Bayesian procedure for Gaussian-assumed density filtering, where the measurement information is gradually included into the given prior estimate is proposed in [27, 28]. The solution is a special case of the framework proposed in this manuscript. A flow for the parameters of the posterior Gaussian density is derived and compared with the state of the art.

A different approach based on a discrete set of bridging densities is pursued in [29, 30]. Here, the likelihood function is decomposed into a product of wider likelihoods that are applied sequentially. After each application and a resulting slight reweighting of the particles, the particles are reapproximated with an equally weighted set. This translates the successive reweighting into a particle motion. For re-approximation, a special type of distance, a so-called permutation-invariant set-distance is employed [31] for comparing two particle sets.

As the online minimization of the distance measure is rather costly, a simplified progression method is used in [32] in the context of Gaussian filters. It is called the Progressive Gaussian Filter (PGF). Here, the re-approximation of weighted particles is performed by replacing them with a pre-calculated set of equally weighted particles corresponding to the same underlying Gaussian density. The original PGF directly works with the measurement equation, no likelihood is required. A more efficient variant for the case that the likelihood function is explicitly available is given in [33]. A semi-analytic variant

of the PGF for measurement equations with a conditionally linear part is derived in [34]. The PGF can easily be parallelized, which is exploited for a GPU-based implementation in [35].

1.2. Contributions

The contributions of this manuscript is a homotopy continuation method called FLUX for approximating complicated probability density functions. FLUX is based on a progressive mechanism for smoothly morphing a given density f_0 into a target density f_1. An approximation for f_0 is already known. The goal is to find an approximation of f_1. The evolution of the densities from f_0 to f_1 is described by distributed ordinary differential equations (DODEs) with an artificial time $\gamma \in [0,1]$. These DODEs are expressed in terms of infinite-dimensional density representations that are impractical for computer implementation and can only be solved in special cases. In the case of a finite-dimensional parametrization, the DODEs can be converted to a system of ordinary differential equations (SODEs), which are solved for $\gamma \in [0,1]$ and return the desired result for $\gamma = 1$. This includes parametric representations such as Gaussians or Gaussian mixtures and nonparametric setups such as sample sets. In the latter case, we obtain a particle flow between the two densities along the artificial time.

FLUX is applied to various density approximation methods. This includes approximating the posterior densities in the measurement update step for state estimation in stochastic nonlinear dynamic systems by gradual inclusion of measurement information. The proposed particle flow method

- is fast,

- can be applied to arbitrary nonlinear systems and is not limited to additive noise,

- allows for target densities that are only known at certain points,

- does not require optimization,

- does not require the solution of partial differential equations, and

- works with standard procedures for solving SODEs.

1.3. Structure of the Paper

This manuscript is structured as follows. In Sec. 2, a rigorous formulation of the pursued approximation problem is given, which includes the most fundamental setup of morphing between two arbitrary densities and also its application to state estimation in stochastic nonlinear dynamic systems. The corresponding distributed ordinary differential equations (DODEs) for describing the evolution of the densities in the infinite-dimensional case are derived in Sec. 3. Exact solutions for special cases are given in Sec. 4. Finite-dimensional approximate solutions are derived for parametric density representations in Sec. 5 and for nonparametric density representations in Sec. 6. Conclusion are given in Sec. 7 followed by an outlook to future work.

2. Problem Formulation

The most fundamental problem we will consider is the approximation of a density $f_1(\underline{x})$ by some parametrized density $f_1(\underline{x}, \underline{\eta}_1)$ with parameter vector $\underline{\eta}_1$. We assume that a somehow related density $f_0(\underline{x})$ exists that is already approximated by a density $f_0(\underline{x}, \underline{\eta}_0)$. Our goal is to find a way to transfer the given approximation $f_0(\underline{x}, \underline{\eta}_0)$ of $f_0(\underline{x})$ to $f_1(\underline{x})$ in order to find $\underline{\eta}_1$.

In the following, we will consider special cases of related densities $f_0(\underline{x})$ and $f_1(\underline{x})$ by estimating posterior densities in the context of stochastic nonlinear dynamic systems. We start with the measurement update or filter step that uses measurements to update the state estimate in Subsec. 2.1. In Subsec. 2.2, we consider the time update or prediction step, where the state estimate is propagated through a nonlinear function and corrupted by noise.

2.1. Measurement Update

We consider a discrete-time system with a hidden state $\underline{x}_k$ at time step k that is observed via a measurement equation

$$\underline{y}_k = \underline{h}_k(\underline{x}_k, \underline{v}_k) \tag{1}$$

with measurement noise $\underline{v}_k$ and a nonlinear measurement function $\underline{h}_k(.,.)$. We assume that a probabilistic description of the measurement equation is given in form of a likelihood function $f_k^L(\underline{y}_k \mid \underline{x}_k)$. We are given a prior density

$$f_k^p(\underline{x}_k) = f(\underline{x}_k \mid \underline{y}_{1:k-1})$$

that summarizes our knowledge about the state based on all past measurements

$$\underline{y}_{1:k-1} = \left[\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_{k-2}, \underline{y}_{k-1} \right]^\top$$

up to the previous time step. We are interested in the posterior density at time step k, which is defined as the density of the state $\underline{x}_k$

$$f_k^e(\underline{x}_k) = f(\underline{x}_k \mid \underline{y}_{1:k})$$

conditioned on all measurements up to the current time step

$$\underline{y}_{1:k} = \left[\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_{k-1}, \underline{y}_k \right]^\top \quad .$$

Via Bayes' law, the desired posterior can be expressed as the normalized product of the prior density before the measurement update and the likelihood describing the measurement equation as

$$f_k^e(\underline{x}_k) = c_k\, f_k^L(\underline{y}_k \mid \underline{x}_k) f_k^p(\underline{x}_k) \quad , \tag{2}$$

where c_k is a normalization constant.

2.1.1. Additive Noise

For additive measurement noise, the measurement equation can be written as

$$\underline{y}_k = \underline{h}_k(\underline{x}_k) + \underline{v}_k \quad . \tag{3}$$

The measurement noise $\underline{v}_k$ is assumed to be zero mean Gaussian with covariance matrix $\mathbf{C}_k^v$ or standard deviation σ_k^v in the scalar case. For the scalar case, the likelihood function is then given as

$$f_k^L(y_k \mid x_k) = \exp\left(-\frac{1}{2} \frac{(y_k - h_k(x_k))^2}{(\sigma_k^v)^2} \right)$$

and in general as

$$f_k^L(\underline{y}_k \mid \underline{x}_k) = \exp\left(-\frac{1}{2} (\underline{y}_k - \underline{h}_k(\underline{x}_k))^\top (\mathbf{C}_k^v)^{-1} (\underline{y}_k - \underline{h}_k(\underline{x}_k)) \right) \quad ,$$

where all multiplicative constants are omitted.

2.1.2. Non-additive Noise

In general, the structure of the measurement equation is not as simple as in the previous subsection. Typically, the noise does not enter additively. We will take a look at two examples: Purely multiplicative noise in Example 2.1 and combined multiplicative and additive noise in Example 2.2.

Example 2.1 (Multiplicative Noise). *As an example, we first consider the case of multiplicative noise for scalar states, i.e., the measurement equation is given by*

$$y_k = x_k \, v_k \quad,$$

with $v_k \sim f_k^v(v_k)$. The conditional density of the measurement y_k given the state x_k and the noise v_k is given by

$$f(y_k \,|\, x_k, v_k) = \delta(y_k - x_k \, v_k) \quad.$$

The conditional density $f(y_k, v_k \,|\, x_k)$ is given by

$$f(y_k, v_k \,|\, x_k) = \delta(y_k - x_k \, v_k) f_k^v(v_k) \quad,$$

so that we obtain

$$f(y_k \,|\, x_k) = \int_{\mathbb{R}} \delta(y_k - x_k \, v_k) f_k^v(v) \, \mathrm{d}v_k \quad,$$

which is equivalent to

$$f(y_k \,|\, x_k) = \int_{\mathbb{R}} \frac{1}{|x_k|} \delta\left(v_k - \frac{y_k}{x_k}\right) f_k^v(v_k) \, \mathrm{d}v \quad.$$

We finally obtain

$$f(y_k \,|\, x_k) = \frac{1}{|x_k|} f_k^v\left(\frac{y_k}{x_k}\right) \quad, \tag{4}$$

which is the likelihood function $f_k^L(x_k)$ for a given y_k. ◀Example▶

We will take a look at a second example that generalizes the one on multiplicative noise. The insights will be used when deriving a progressive update for the multiplicative noise case.

Example 2.2 (Multiplicative and Additive Noise). *We now consider the case of a measurement corrupted by multiplicative and additive noise as*

$$y_k = x_k \, v_k + w_k \quad,$$

with $v_k, w_k \sim f_k^{vw}(v_k, w_k)$. The conditional density of the measurement y_k given the state x_k and the noise terms v_k and w_k is given by

$$f(y_k \,|\, x_k, v_k, w_k) = \delta(y_k - x_k \, v_k - w_k) \quad.$$

The conditional density $f(y_k, v_k, w_k \,|\, x_k)$ is given by

$$f(y_k, v_k, w_k \,|\, x_k) = \delta(y_k - x_k \, v_k - w_k) f_k^{vw}(v_k, w_k) \quad.$$

Marginalization gives

$$
\begin{aligned}
f(y_k \,|\, x_k) &= \int_{\mathbb{R}} \int_{\mathbb{R}} f(y_k, v_k, w_k \,|\, x_k) \, \mathrm{d}v_k \, \mathrm{d}w_k \\
&= \int_{\mathbb{R}} \int_{\mathbb{R}} \delta(y_k - x_k \, v_k - w_k) f_k^{vw}(v_k, w_k) \, \mathrm{d}v_k \, \mathrm{d}w_k \\
&= \int_{\mathbb{R}} f_k^{vw}(v_k, y_k - x_k \, v_k) \, \mathrm{d}v_k \quad.
\end{aligned}
$$

For v_k and w_k independent, i.e., $f_k^{vw}(v_k, w_k) = f_k^v(v_k) \, f_k^w(w_k)$, we obtain

$$f(y_k \,|\, x_k) = \int_{\mathbb{R}} f_k^v(v_k) \, f_k^w(y_k - x_k \, v_k) \, \mathrm{d}v_k \quad.$$

5

For $f_k^v(v_k)$ and $f_k^w(w_k)$ Gaussian densities $f_k^v(v_k) \sim \mathcal{N}(v_k, m_k^v, \sigma_k^v)$ and $f_k^w(w_k) \sim \mathcal{N}(w_k, m_k^w, \sigma_k^w)$, the integral can be evaluated in closed-form as

$$f(y_k \,|\, x_k) = \frac{1}{\sqrt{2\pi}\sqrt{(\sigma_k^v)^2 x_k^2 + (\sigma_k^w)^2}} \exp\left\{ -\frac{1}{2} \frac{(y_k - m_k^v x_k - m_k^w)^2}{(\sigma_k^v)^2 x_k^2 + (\sigma_k^w)^2} \right\} \quad .$$

For $m_k^v = 0$ and $m_k^w = 0$ this gives

$$f(y_k \,|\, x_k) = \frac{1}{\sqrt{2\pi}\sqrt{(\sigma_k^v)^2 x_k^2 + (\sigma_k^w)^2}} \exp\left\{ -\frac{1}{2} \frac{y_k^2}{(\sigma_k^v)^2 x_k^2 + (\sigma_k^w)^2} \right\} \quad .$$

When the additive noise w_k goes to zero, we obtain

$$f(y_k \,|\, x_k) = \frac{1}{\sqrt{2\pi}(\sigma_k^v)^2 |x_k|} \exp\left\{ -\frac{1}{2} \frac{y_k^2}{(\sigma_k^v)^2 v \, x_k^2} \right\} \quad ,$$

which is equivalent to (4) in Example 2.1 for Gaussian noise $f_k^v(.)$. When the multiplicative noise vanishes, which can be expressed as $f_k^v(v_k) = \delta(v_k - 1)$ or $m_k^v = 1$ and $\sigma_k^v \to 0$, we obtain

$$f(y_k \,|\, x_k) = \frac{1}{\sqrt{2\pi}\sigma_k^w} \exp\left\{ -\frac{1}{2} \frac{(y_k - x_k)^2}{(\sigma_k^w)^2} \right\} \quad ,$$

which is the result that is also directly obtained for the purely additive case. ◀Example▶

2.2. Time Update

So far, we looked at a single measurement update, where a given measurement is used to update a prior estimate of the system state. Now we consider the propagation of a given state estimate through a discrete-time nonlinear stochastic system

$$\underline{x}_{k+1} = \underline{a}_k(\underline{x}_k, \underline{w}_k)$$

with time step k, state $\underline{x}_k$, system noise $\underline{w}_k$, and nonlinear system function $\underline{a}_k(.,.)$. In analogy to the likelihood function as a probabilistic description of the measurement equation, we will use a probabilistic representation of the system equation. This so-called transition density is given by

$$f(\underline{x}_{k+1} \,|\, \underline{x}_k) \quad .$$

It is calculated based on the system equation in a similar way as the likelihood function is determined based on the measurement equation.

3. Zakai-type Distributed Ordinary Differential Equations in Artificial Time

In this section, we will derive distributed ordinary differential equations (DODEs) that smoothly morph one density into another. Here, we will focus on infinite-dimensional representations, i.e., function spaces, with corresponding solutions derived in Sec. 4 for some special cases. Finite-dimensional spaces will be considered in Sec. 5 and Sec. 6. Parametric solutions are derived in Sec. 5, while Sec. 6 covers non-parametric solutions.

In the following, we omit the time index k for simplicity.

3.1. Key Idea

The most general setup is as follows. We are given two densities $f_0(\underline{x})$ and $f_1(\underline{x})$. The first density $f_0(\underline{x})$ is not necessarily known, but an approximation $f_0(\underline{x}, \underline{\eta}_0)$ is given, where $\underline{\eta}_0$ is a parameter vector. The second density, the target, may only be implicitly known or only be given at certain points. Our goal is to find an approximation $f_1(\underline{x}, \underline{\eta}_1)$. Finding the parameter vector $\underline{\eta}_1$ typically is a complicated optimization problem.

The key idea for efficiently obtaining $\underline{\eta}_1$ is to first find a smooth progression for morphing $f_0(\underline{x})$ into $f_1(\underline{x})$ depending on an artificial time parameter $\gamma \in [0, 1]$, i.e.,

$$f_\gamma(\underline{x}) = \begin{cases} f_0(\underline{x}) & \gamma = 0 \\ \text{nice interpolation} & \gamma \in (0, 1) \\ f_1(\underline{x}) & \gamma = 1 \end{cases} .$$

Second, this progression is expressed as a DODE

$$\dot{f}_\gamma(\underline{x}) = \frac{\partial f_\gamma(\underline{x})}{\partial \gamma} = a_\gamma(\underline{x}) f_\gamma(\underline{x}) + b_\gamma(\underline{x})$$

depending on the artificial time $\gamma \in [0, 1]$ and the state $\underline{x}$. $a_\gamma(\underline{x})$ and $b_\gamma(\underline{x})$ are suitable functions, that will be derived in the following for specific problems. The initial condition is $f_{\gamma=0}(\underline{x}) = f_0(\underline{x})$ and for $\gamma = 1$, we obtain the desired density $f_1(\underline{x})$. Third, this DODE is transferred to the approximate parametrized densities $f_0(\underline{x}, \underline{\eta}_0)$ and $f_1(\underline{x}, \underline{\eta}_1)$.

Remark 3.1. *It is important to note the following two properties of the DODE:*

- *The derived evolution equation is* not *a partial differential equation, but rather an ordinary differential equation for all values of the state $\underline{x}$. Hence, it is called a distributed ordinary differential equation (DODE).*

- *We consider the evolution of* unnormalized *densities. This is the reason, why we call them Zakai-type DODEs.*

These two properties make the solution for $f_\gamma(\underline{x})$, $\gamma \in [0, 1]$ much simpler compared to state of the art approaches. ◀**Remark**▶

Fourth, by averaging the DODE resulting from inserting the approximate densities over the domain $\underline{x}$, we finally obtain a system of ordinary differential equations for the parameter vector

$$\dot{\underline{\eta}}_\gamma = \mathbf{M}(\underline{\eta}_\gamma) \underline{\eta}_\gamma$$

with

$$\underline{\eta}_\gamma = \begin{cases} \underline{\eta}_0 & \gamma = 0 \\ \underline{\eta}_1 & \gamma = 1 \end{cases} .$$

$\mathbf{M}(\underline{\eta}_\gamma)$ is a matrix depending on the functions $a_\gamma(\underline{x})$ and $b_\gamma(\underline{x})$ and the selected approximate representation $f_\gamma(\underline{x}, \underline{\eta}_\gamma)$.

3.2. DODE for Morphing from One Density to Another Density

The simplest progression to move from f_0 to f_1 is given by

$$f_\gamma(\underline{x}) = (1 - g_\gamma) f_0(\underline{x}) + g_\gamma f_1(\underline{x}) = f_0(\underline{x}) + g_\gamma (f_1(\underline{x}) - f_0(\underline{x})) ,$$

with $g_\gamma \in [0, 1]$, i.e., $g_0 = 0$ and $g_1 = 1$, g_γ continuous, and g_γ strictly increasing. Taking the derivative with respect to γ gives

$$\dot{f}_\gamma(\underline{x}) = \dot{g}_\gamma (f_1(\underline{x}) - f_0(\underline{x}))$$

7

or with

$$f_1(\underline{x}) - f_0(\underline{x}) = \frac{1}{g_\gamma}\left(f_\gamma(\underline{x}) - f_0(\underline{x})\right)$$

the DODE

$$\dot{f}_\gamma(\underline{x}) = \frac{\dot{g}_\gamma}{g_\gamma}\left(f_\gamma(\underline{x}) - f_0(\underline{x})\right) \ ,$$

where we have to take care of the singularity due to g_γ when $\gamma = 0$.

Remark 3.2. *The function g_γ determines the rate of change in the densities when γ varies from 0 to 1. Of course, we could use $g_\gamma = \gamma$ when the expected change is relatively homogeneous over γ anyway. However, when foreseeable variations in the rate of change occur, an appropriate selection of g_γ can simplify the life of the DODE solver.* ◀Remark▶

3.3. DODE for Measurement Update

In the case of a measurement update according to (2), we set $f_0(\underline{x}) = f_p(\underline{x})$ and $f_1(\underline{x}) = f_e(\underline{x})$. Omitting the normalization constant in (2), the relation between the two densities is given by

$$f_e(\underline{x}) = f_L(\underline{x})\, f_p(\underline{x}) \ .$$

We now introduce a progressive likelihood function $f_L(\underline{x}, \gamma)$ with the property

$$f_L(\underline{x}, \gamma) = \begin{cases} 1 & \gamma = 0 \\ f_L(\underline{x}) & \gamma = 1 \end{cases}$$

and we obtain

$$f_e(\underline{x}, \gamma) = f_L(\underline{x}, \gamma)\, f_p(\underline{x}) \ .$$

Taking the derivative with respect to γ gives

$$\dot{f}_e(\underline{x}, \gamma) = \dot{f}_L(\underline{x}, \gamma)\, f_p(\underline{x})$$

or

$$\dot{f}_e(\underline{x}, \gamma) = \frac{\dot{f}_L(\underline{x}, \gamma)}{f_L(\underline{x}, \gamma)}\, f_e(\underline{x}, \gamma) \ ,$$

which is the desired DODE.

A straightforward progression is

$$f_L(\underline{x}, \gamma) = [f_L(\underline{x})]^{g(\gamma)} \ .$$

Its derivative is

$$\begin{aligned}
\dot{f}_L(\underline{x}, \gamma) &= \frac{\partial}{\partial \gamma}\left\{\left[\exp\left(\log\left(f_L(\underline{x})\right)\right)\right]^{g(\gamma)}\right\} \\
&= \frac{\partial}{\partial \gamma}\left\{\exp\left(g(\gamma)\log\left(f_L(\underline{x})\right)\right)\right\} \\
&= \log\left(f_L(\underline{x})\right)\dot{g}(\gamma)\, f_L(\underline{x}) \ .
\end{aligned}$$

The DODE for this progression is then

$$\dot{f}_e(\underline{x}, \gamma) = \log\left(f_L(\underline{x}, \gamma)\right)\dot{g}(\gamma)\, f_e(\underline{x}, \gamma) \ .$$

For exponential families, we obtain simpler expressions due to the logarithm taken above. For the special case of additive Gaussian measurement noise according to Subsec. 2.1.1 and $g(\gamma) = \gamma$, the likelihood function is

$$f_L(\underline{x}, \gamma) = \exp\left(-\gamma\frac{1}{2}(\underline{y} - \underline{h}(\underline{x}))^\top \mathbf{C}_v^{-1}(\underline{y} - \underline{h}(\underline{x}))\right) \ ,$$

and in the scalar case

$$f_L(x, \gamma) = \exp\left(-\gamma \frac{1}{2} \frac{(y - h(x))^2}{\sigma_v^2}\right) \quad .$$

The corresponding DODE is

$$\dot{f}_e(\underline{x}, \gamma) = -\frac{1}{2}(\underline{y} - \underline{h}(\underline{x}))^\top \mathbf{C}_v^{-1}(\underline{y} - \underline{h}(\underline{x})) \, f_e(\underline{x}, \gamma)$$

and in the scalar case

$$\dot{f}_e(x, \gamma) = -\frac{1}{2} \frac{[y - h(x)]^2}{\sigma_v^2} \, f_e(x, \gamma) \quad . \tag{5}$$

4. Zakai Equation: Exact Solution

The distributed ordinary differential equation (DODE) in (5) can be solved exactly in a few simple (but interesting) cases. In general, however, approximate solution methods have to be employed. For the special case of a linear measurement equation, the solution procedure is shown in the next example. Of course, we obtain the expected result for the posterior density that in this case could have been easily obtained by directly solving (2).

Example 4.1. *We consider a prior zero-mean Gaussian density*

$$f_e(x, \gamma = 0) = f_p(x) = \mathcal{N}(x, 0, \sigma_p^2) = \frac{1}{\sqrt{2\pi}\sigma_p} \exp\left(-\frac{1}{2} \frac{x^2}{\sigma_p^2}\right)$$

with standard deviation σ_p. The measurement equation is assumed to be simply

$$y = x + v \quad ,$$

with measurement y and measurement noise $v \sim \mathcal{N}(v, 0, \sigma_v)$, which corresponds to the likelihood function

$$f_L(x) = \exp\left(-\frac{1}{2} \frac{(y - x)^2}{\sigma_v^2}\right) \quad ,$$

where we set $y = 0$ for simplicity. According to (5), the DODE is given by

$$\dot{f}_e(x, \gamma) = -\frac{1}{2} \frac{x^2}{\sigma_v^2} \, f_e(x, \gamma) \quad . \tag{6}$$

For the solution of the DODE, we use the ansatz

$$\bar{f}_e(x, \gamma) = k_1 \, \exp(k_2 \, \gamma \, x^2)$$

with derivative with respect to γ given by

$$\dot{\bar{f}}_e(x, \gamma) = k_1 \, k_2 \, x^2 \, \exp(k_2 \, \gamma \, x^2) = k_2 \, x^2 \, f_e(x, \gamma) \quad .$$

Inserting $\bar{f}_e(x, \gamma)$ and $\dot{\bar{f}}_e(x, \gamma)$ into the DODE gives

$$k_2 \, x^2 \, f_e(x, \gamma) = -\frac{1}{2} \frac{x^2}{\sigma_v^2} f_e(x, \gamma)$$

and it follows that

$$k_2 = -\frac{1}{2} \frac{1}{\sigma_v^2} \quad .$$

Using the initial condition

$$\dot{\bar{f}}_e(x, \gamma = 0) = k_1 = f_e(x, \gamma = 0) = \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{1}{2}\frac{x^2}{\sigma_p^2}\right)$$

gives

$$\bar{f}_e(x, \gamma) = \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{1}{2}\frac{x^2}{\sigma_p^2}\right) \exp\left(-\gamma\frac{1}{2}\frac{x^2}{\sigma_v^2}\right)$$

$$= \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_p^2} + \gamma\frac{x^2}{\sigma_v^2}\right)\right) \quad .$$

This can be written as

$$\bar{f}_e(x, \gamma) = \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{1}{2}\frac{x^2}{\sigma_e^2}\right) \tag{7}$$

with

$$\sigma_e^2 = \frac{1}{\frac{1}{\sigma_p^2} + \gamma\frac{1}{\sigma_v^2}} \quad .$$

Obviously, we get the same result as solving (2) directly for this special case of a linear system. Please note that the posterior density in (7) is unnormalized *(in this case, normalized with respect to the prior density) as the DODE in (6) describes the evolution of the unnormalized density starting at the (in this case) normalized prior.* ◄**Example**►

5. Parametric Solution of Zakai Equation

For simplifying the exposition, we begin with the scalar DODE (5) repeated here for convenience

$$\dot{f}_e(x, \gamma) = -\frac{1}{2}\frac{[y - h(x)]^2}{\sigma_v^2} f_e(x, \gamma) \quad .$$

For solving this DODE, we plug in a parametric representation $f_e(x, \gamma) = f_e(x, \underline{\eta}_e(\gamma))$ with parameter vector $\underline{\eta}_e(\gamma)$, which gives

$$\dot{f}_e(x, \underline{\eta}_e(\gamma)) = -\frac{1}{2}\frac{[y - h(x)]^2}{\sigma_v^2} f_e(x, \underline{\eta}_e(\gamma)) \quad . \tag{8}$$

In order to calculate the desired parameter vector $\underline{\eta}_e(\gamma)$ for $\gamma = 1$, the DODE has to be solved for $\gamma \in [0, 1]$ and for all states x simultaneously. However, for a given parametric representation, the DODE typically cannot be exactly solved over all states x. Hence, we have to find a parameter vector $\underline{\eta}_e(\gamma)$ in such a way that some average distance between the left-hand side and the right-hand side of (8) is minimized for all γ. We will pursue two options for minimizing an average distance: (1) a continuous version based on the squared integral distance and (2) a discrete version based on distances at collocation points. Both options will be used for different parametric representations.

5.1. Parametric Solution of Zakai Equation: Gaussian Posteriors

As a specific parametric representation of the posterior density, we assume that the state pdf can be represented by an (unnormalized) posterior Gaussian density, i.e.,

$$f_e(x, \underline{\eta}_e(\gamma)) = k_e(\gamma) \exp\left(-\frac{1}{2}\frac{[x - m_e(\gamma)]^2}{\sigma_e^2(\gamma)}\right) \quad , \tag{9}$$

with

$$\underline{\eta}_e(\gamma) = \begin{bmatrix} k_e(\gamma) & m_e(\gamma) & \sigma_e(\gamma) \end{bmatrix}^\top$$

depending on the artificial time γ.

10

Remark 5.1. *The factor $k_e(\gamma)$ is not used for normalizing $f_e(x, \underline{\eta}_e(\gamma))$. For the start of the progression, i.e., $\gamma = 0$, it can be initialized with any convenient value such as 1 or the normalization constant of the prior density. During the progression, it is then used to take care of the change in probability mass due to the multiplication with the likelihood function. When the progression reaches $\gamma = 1$, $k_e(\gamma)$ is generally discarded and the density $f_e(x, \underline{\eta}_e(\gamma))$ properly normalized.* ◀**Remark**▶

The derivative of $f_e(x, \underline{\eta}_e(\gamma))$ with respect to γ is given by

$$\dot{f}_e(x, \underline{\eta}_e(\gamma)) = \left(\frac{1}{k_e(\gamma)} \dot{k}_e(\gamma) + \frac{x - m_e(\gamma)}{\sigma_e^2(\gamma)} \dot{m}_e(\gamma) + \frac{[x - m_e(\gamma)]^2}{\sigma_e^3(\gamma)} \dot{\sigma}_e(\gamma) \right) f_e(x, \underline{\eta}_e(\gamma)) \ . \tag{10}$$

Plugging (10) into (8) results in a parametric DODE

$$\left(\frac{1}{k_e(\gamma)} \dot{k}_e(\gamma) + \frac{x - m_e(\gamma)}{\sigma_e^2(\gamma)} \dot{m}_e(\gamma) + \frac{[x - m_e(\gamma)]^2}{\sigma_e^3(\gamma)} \dot{\sigma}_e(\gamma) \right) f_e(x, \underline{\eta}_e(\gamma)) = -\frac{1}{2} \frac{[y - h(x)]^2}{\sigma_v^2} f_e(x, \underline{\eta}_e(\gamma)) \ . \tag{11}$$

Our goal is to solve the parametric DODE (11) for $\gamma \in [0, 1]$ in order to obtain the values of $k_e(\gamma)$, $m_e(\gamma)$, and $\sigma_e(\gamma)$ for $\gamma = 1$.

5.1.1. Squared Integral Solution

For solving the parametric DODE (11), we first employ a squared integral distance measure

$$D = \int_{\mathbb{R}} \left(\underbrace{\frac{1}{k_e(\gamma)} \dot{k}_e(\gamma) + \frac{x - m_e(\gamma)}{\sigma_e^2(\gamma)} \dot{m}_e(\gamma) + \frac{[x - m_e(\gamma)]^2}{\sigma_e^3(\gamma)} \dot{\sigma}_e(\gamma) + \frac{1}{2} \frac{[y - h(x)]^2}{\sigma_v^2}}_{\Delta(x, \gamma)} \right)^2 f_e^2(x, \underline{\eta}_e(\gamma)) \, \mathrm{d}x \ . \tag{12}$$

Taking the derivative of D with respect to $\dot{k}_e(\gamma)$, $\dot{m}_e(\gamma)$, and $\dot{\sigma}_e(\gamma)$ gives

$$D_k = \frac{\partial D}{\partial \dot{k}_e(\gamma)} = 2 \int_{\mathbb{R}} \Delta(x, \gamma) \frac{1}{k_e(\gamma)} f_e^2(x, \underline{\eta}_e(\gamma)) \, \mathrm{d}x \ ,$$

$$D_m = \frac{\partial D}{\partial \dot{m}_e(\gamma)} = 2 \int_{\mathbb{R}} \Delta(x, \gamma) \frac{x - m_e(\gamma)}{\sigma_e^2(\gamma)} f_e^2(x, \underline{\eta}_e(\gamma)) \, \mathrm{d}x \ ,$$

$$D_\sigma = \frac{\partial D}{\partial \dot{\sigma}_e(\gamma)} = 2 \int_{\mathbb{R}} \Delta(x, \gamma) \frac{(x - m_e(\gamma))^2}{\sigma_e^3(\gamma)} f_e^2(x, \underline{\eta}_e(\gamma)) \, \mathrm{d}x \ .$$

The parameters $k_e(\gamma)$, $m_e(\gamma)$, and $\sigma_e(\gamma)$ corresponding to the minimum of D in (12) for all γ is obtained by setting D_k, D_m, and D_σ to zero, which gives

$$\mathbf{P}\big(\underline{\eta}_e(\gamma)\big) \cdot \underline{\dot{\eta}}_e(\gamma) = \underline{q}(\underline{\eta}_e(\gamma)) \tag{13}$$

with

$$\underline{\dot{\eta}}_e(\gamma) = \left[\dot{k}_e(\gamma), \dot{m}_e(\gamma), \dot{\sigma}_e(\gamma) \right]^\top$$

and

$$\mathbf{P}\big(\underline{\eta}_e(\gamma)\big) = \begin{bmatrix} I\left(\frac{1}{k_e^2(\gamma)}\right) & I\left(\frac{1}{k_e(\gamma)} \frac{x - m_e(\gamma)}{\sigma_e^2(\gamma)}\right) & I\left(\frac{1}{k_e(\gamma)} \frac{(x - m_e(\gamma))^2}{\sigma_e^3(\gamma)}\right) \\[2ex] I\left(\frac{1}{k_e(\gamma)} \frac{x - m(\gamma)}{\sigma_e^2(\gamma)}\right) & I\left(\frac{(x - m_e(\gamma))^2}{\sigma_e^4(\gamma)}\right) & I\left(\frac{(x - m_e(\gamma))^3}{\sigma_e^5(\gamma)}\right) \\[2ex] I\left(\frac{1}{k_e(\gamma)} \frac{(x - m_e(\gamma))^2}{\sigma_e^3(\gamma)}\right) & I\left(\frac{(x - m_e(\gamma))^3}{\sigma_e^5(\gamma)}\right) & I\left(\frac{(x - m_e(\gamma))^4}{\sigma_e^6(\gamma)}\right) \end{bmatrix} \ , \tag{14}$$

$$n = 0: \ I\left((x - m_e(\gamma))^0\right) = \int_{\mathbb{R}} f_e^2(\underline{x}, \gamma) \qquad\qquad = \sqrt{\pi}\, k_e^2(\gamma)\, \sigma_e(\gamma) \ ,$$

$$n = 1: \ I\left((x - m_e(\gamma))^1\right) = \int_{\mathbb{R}} (x - m_e(\gamma))\, f_e^2(x, \underline{\eta}_e(\gamma)) \ = 0 \ ,$$

$$n = 2: \ I\left((x - m_e(\gamma))^2\right) = \int_{\mathbb{R}} (x - m_e(\gamma))^2\, f_e^2(x, \underline{\eta}_e(\gamma)) = \frac{1}{2}\sqrt{\pi}\, k_e^2(\gamma)\, \sigma^3(\gamma) \ ,$$

$$n = 3: \ I\left((x - m_e(\gamma))^3\right) = \int_{\mathbb{R}} (x - m_e(\gamma))^3\, f_e^2(x, \underline{\eta}_e(\gamma)) = 0 \ ,$$

$$n = 4: \ I\left((x - m_e(\gamma))^4\right) = \int_{\mathbb{R}} (x - m_e(\gamma))^4\, f_e^2(x, \underline{\eta}_e(\gamma)) = \frac{3}{4}\sqrt{\pi}\, k_e^2(\gamma)\, \sigma_e^5(\gamma) \ .$$

Figure 1: Special cases $n = 0, \ldots, 4$ for the integral $I\left((x - m_e(\gamma))^n\right)$ in (16).

where $I(.)$ is defined as

$$I\left(\mathrm{g}(x)\right) = \int_{\mathbb{R}} \mathrm{g}(x)\, f_e^2(x, \underline{\eta}_e(\gamma))\, \mathrm{d}x \ .$$

The right-hand side of (13) is given by

$$\underline{q}(\underline{\eta}_e(\gamma)) = -\frac{1}{2}\begin{bmatrix} \int_{\mathbb{R}} \dfrac{[y - h(x)]^2}{\sigma_v^2} \dfrac{1}{k_e(\gamma)}\, f_e^2(x, \underline{\eta}_e(\gamma))\, \mathrm{d}x \\[2em] \int_{\mathbb{R}} \dfrac{[y - h(x)]^2}{\sigma_v^2} \dfrac{x - m_e(\gamma)}{\sigma_e^2(\gamma)}\, f_e^2(x, \underline{\eta}_e(\gamma))\, \mathrm{d}x \\[2em] \int_{\mathbb{R}} \dfrac{[y - h(x)]^2}{\sigma_v^2} \dfrac{(x - m_e(\gamma))^2}{\sigma_e^3(\gamma)}\, f_e^2(x, \underline{\eta}_e(\gamma))\, \mathrm{d}x \end{bmatrix} . \tag{15}$$

For solving the integral expressions in (14), we use the following moment expressions

$$I\left((x - m_e(\gamma))^n\right) = \int_{\mathbb{R}} (x - m_e(\gamma))^n\, f_e^2(x, \underline{\eta}_e(\gamma))\, \mathrm{d}x = \begin{cases} \sqrt{\pi}\, k_e^2(\gamma)\, \dfrac{(n-1)(n-3)\cdots 1}{2^{n/2}} \sigma^{n+1}(\gamma) & \text{for } n \text{ even} \\ 0 & \text{for } n \text{ odd} \end{cases} . \tag{16}$$

For the cases $n = 0, \ldots, 4$, expressions are given in Fig. 1. As a result, we obtain

$$\mathbf{P}\left(\underline{\eta}_e(\gamma)\right) = \sqrt{\pi}\begin{bmatrix} \sigma_e(\gamma) & 0 & \frac{1}{2}k_e(\gamma) \\[1em] 0 & \frac{1}{2}\frac{k_e^2(\gamma)}{\sigma_e(\gamma)} & 0 \\[1em] \frac{1}{2}k_e(\gamma) & 0 & \frac{3}{4}\frac{k_e^2(\gamma)}{\sigma_e(\gamma)} \end{bmatrix} .$$

Analytic Solution for $\underline{q}(\underline{\eta}_e(\gamma))$ in (15): In many interesting cases, an analytic solution for $\underline{q}(\underline{\eta}_e(\gamma))$ can be found. The following example gives the solution for the case of a linear system. This serves as a sanity check, as in this case, the exact solution is known.

Example 5.1. *We consider a scalar linear measurement equation, i.e., $h(x)$ in (3) is now given as $h(x) = H\,x$. In this case, we obtain*

$$\underline{q}(\underline{\eta}_e(\gamma)) = \frac{\sqrt{\pi}}{2\,\sigma_v^2} \begin{bmatrix} -\dfrac{1}{2}k_e(\gamma)\,\sigma_e(\gamma)\,\left\{ H^2\sigma_e^2(\gamma) + 2(y - H\,m_e(\gamma))^2 \right\} \\[2ex] H\,k_e^2(\gamma)\,\sigma_e(\gamma)\,(y - H\,m_e(\gamma)) \\[2ex] -\dfrac{1}{4}k_e^2(\gamma)\,\left\{ 3H^2\sigma_e^2(\gamma) + 2(y - H\,m_e(\gamma))^2 \right\} \end{bmatrix}$$

as the right-hand side of (13).   ◀**Example**▶

For solving the ODE (13), we use a standard ODE solver. We avoid the explicit inversion of the matrix $\mathbf{P}(\underline{\eta}(\gamma))$ by interpreting (13) as a linear equation that is solved for $\dot{\underline{\eta}}(\gamma)$. The ODE solver is accelerated by using the Jacobian matrix of $\underline{q}(\underline{\eta}_e(\gamma))$ given by

$$J(\underline{\eta}_e(\gamma)) = \frac{\partial \underline{q}^\top(\underline{\eta}_e(\gamma))}{\partial \underline{\eta}_e(\gamma)}\ .$$

There are other interesting nonlinear functions $h(x)$, where the analytic solution for $\underline{q}(\underline{\eta}_e(\gamma))$ in (15) is possible. This includes polynomials, trigonometric functions, and exponential functions.

Numeric Solution for $\underline{q}(\underline{\eta}_e(\gamma))$ in (15): When an analytic solution is not possible, (15) has to be evaluated numerically. An elegant quadrature rule is obtained by replacing $f_e^2(x, \underline{\eta}_e(\gamma))$ by its Dirac mixture approximation given by

$$f_e^2(x, \gamma) \approx \sum_{i=0}^{N} w_i(\gamma)\,\delta(x - x_i(\gamma))\ .$$

Inserting this approximation into (15) gives

$$\underline{q}\left(\underline{\eta}(\gamma)\right) \approx -\frac{1}{2}\sum_{i=0}^{N} w_i(\gamma)\,\frac{\left[y - h(x_i(\gamma))\right]^2}{\sigma_v^2} \begin{bmatrix} \dfrac{1}{k_e(\gamma)} \\[2ex] \dfrac{x_i(\gamma) - m_e(\gamma)}{\sigma_e^2(\gamma)} \\[2ex] \dfrac{(x_i(\gamma) - m_e(\gamma))^2}{\sigma_e^3(\gamma)} \end{bmatrix}\ .$$

Remark 5.2. *Of course, we could now ask ourselves why this numerical procedure is simpler than just performing a numeric integration of the basic update equation (2) for calculating the desired posterior moments. First, the square of the progressive posterior $f_e^2(x, \underline{\eta}_e(\gamma))$ is always narrower than the prior $f_p(x)$ even for $\gamma = 0$ and gets narrower with increasing γ. Hence, we have to integrate over a smaller region. Second, we do not have to integrate over the likelihood as in (2), we only have to calculate (nonlinear) moments of $f_e^2(x, \underline{\eta}_e(\gamma))$. Third, this procedure also works for Gaussian Mixture (GM) posteriors, not only for Gaussian posteriors. For GM posteriors, moments of (2) would not be helpful for obtaining the posterior mixture parameters.*   ◀**Remark**▶

5.1.2. Solution at Collocation Points

In Subsec. 5.1.1, the distributed ordinary differential equation (11) was solved "on average" over the state domain by minimizing a squared integral distance between its left-hand side and its right-hand side for calculating $\dot{\underline{\eta}}_e(\gamma)$. Here, we will solve the DODE (11) at selected discrete points only. There are several ways of selecting these points. The natural solution is to use a Dirac mixture approximation of $f_e(x, \underline{\eta}_e(\gamma))$

$$f_e(x, \underline{\eta}_e(\gamma)) \approx \sum_{i=0}^{C} w_i(\gamma)\,\delta(x - x_i(\gamma))\ ,$$

13

with $x_i(\gamma)$ the collocation points for $i = 1, \ldots, C$. The weights $w_i(\gamma)$ are typically selected to be equal. This automatically places more points where the density is large, putting a higher priority to large density regions.

By plugging these collocation points into (11), we obtain

$$\frac{1}{k_e(\gamma)} \dot{k}_e(\gamma) + \frac{x_i(\gamma) - m_e(\gamma)}{\sigma_e^2(\gamma)} \dot{m}_e(\gamma) + \frac{[x_i(\gamma) - m_e(\gamma)]^2}{\sigma_e^3(\gamma)} \dot{\sigma}_e(\gamma) = -\frac{1}{2} \frac{[y - h(x_i(\gamma))]^2}{\sigma_v^2}$$

for $i = 1, \ldots, C$. In vector-matrix-form, in analogy to (13), this gives

$$\mathbf{P}(\underline{\eta}_e(\gamma)) \cdot \dot{\underline{\eta}}_e(\gamma) = \underline{q}(\underline{\eta}_e(\gamma))$$

with

$$\mathbf{P}(\underline{\eta}_e(\gamma)) = \begin{bmatrix} \frac{1}{k_e(\gamma)} & \frac{x_1(\gamma) - m_e(\gamma)}{\sigma_e^2(\gamma)} & \frac{[x_1(\gamma) - m_e(\gamma)]^2}{\sigma_e^3(\gamma)} \\ \frac{1}{k_e(\gamma)} & \frac{x_2(\gamma) - m_e(\gamma)}{\sigma_e^2(\gamma)} & \frac{[x_2(\gamma) - m_e(\gamma)]^2}{\sigma_e^3(\gamma)} \\ \vdots & \vdots & \vdots \\ \frac{1}{k_e(\gamma)} & \frac{x_i(\gamma) - m_e(\gamma)}{\sigma_e^2(\gamma)} & \frac{[x_i(\gamma) - m_e(\gamma)]^2}{\sigma_e^3(\gamma)} \\ \vdots & \vdots & \vdots \\ \frac{1}{k_e(\gamma)} & \frac{x_C(\gamma) - m_e(\gamma)}{\sigma_e^2(\gamma)} & \frac{[x_C(\gamma) - m_e(\gamma)]^2}{\sigma_e^3(\gamma)} \end{bmatrix}$$

and

$$\underline{q}(\underline{\eta}_e(\gamma)) = -\frac{1}{2} \frac{1}{\sigma_v^2} \begin{bmatrix} [y - h(x_1(\gamma))]^2 \\ [y - h(x_2(\gamma))]^2 \\ \vdots \\ [y - h(x_i(\gamma))]^2 \\ \vdots \\ [y - h(x_C(\gamma))]^2 \end{bmatrix} \ .$$

6. Nonparametric Solution of Zakai Equation

Here, we extend the previous ideas to the case of nonparametric solutions. By "nonparametric" we mean that we directly use a set of samples that is updated during the measurement step. No fitting of these samples to a parametric density is performed.

We are given a set of L_p prior sample vectors $\underline{x}_{p,i}$, $i = 1, \ldots, L_p$ arranged in a matrix according to

$$\mathbf{X}_p = \begin{bmatrix} \underline{x}_{p,1}, \underline{x}_{p,2}, \ldots, \underline{x}_{p,i}, \ldots, \underline{x}_{p,L_p} \end{bmatrix} \ .$$

Our goal is to find a posterior set of L_e sample vectors $\underline{x}_{e,i}(\gamma)$, $i = 1, \ldots, L_e$ with

$$\mathbf{X}_e(\gamma) = \begin{bmatrix} \underline{x}_1(\gamma), \underline{x}_2(\gamma), \ldots, \underline{x}_i(\gamma), \ldots, \underline{x}_{L_e}(\gamma) \end{bmatrix} \ .$$

For simplicity, we start with the case of equally weighted samples in the prior and the posterior set $\mathbf{X}_p$ and $\mathbf{X}_e(\gamma)$. We also assume that the number of components does not change during the update, so $L = L_p = L_e$. Furthermore, we assume a scalar state domain.

The key idea is to reconstruct the values of the density underlying the given samples at the sample locations. A simple density-spacings estimator is used for that purpose. An estimate of the density underlying the sample set $\mathbf{X}_e(\gamma)$ is given by

$$f_e(x_i, \gamma) = \frac{1}{2} \frac{w_i}{|x_i(\gamma) - x_j(\gamma)|} \ ,$$

where $w_i = 1/L$ and j is the index of the nearest neighbor sample to sample i. We write $j = \mathrm{NN}(i)$.

Remark 6.1. *Considering the density only at the given L sample locations would directly give us L ODEs for the desired posterior locations. However, even in the one-dimensional case we need $L + 1$ equations as we have to propagate the L locations plus some factor that takes care of the fact that we consider unnormalized densities.*

Of course, in higher-dimensional state spaces, say in D dimensions, $L \cdot D$ parameters are required for representing L samples plus one parameter for the factor. As in the scalar case, considering density values at the component locations still only gives us L equations for, now, $L \cdot D + 1$ parameters.

In order to obtain more equations than unknowns, we perform first a continuous interpolation between the density values leading to a DODE that is continuous over the state space $\mathcal{X}$. Second, this continuous DODE is evaluated at N carefully selected collocation points with $N > L \cdot D + 1$. resulting in a system of N ordinary differential equations.   ◄Remark►

For interpolation, we use a simple Gaussian kernel

$$f_{e,i}(x, \gamma) = f_e(x_i, \gamma) \exp\left(-\frac{1}{2}\frac{(x - x_i(\gamma))^2}{\sigma_i^2(\gamma)}\right) \ ,$$

where $\sigma_i(\gamma)$ is chosen in such a way that the integral over $f_{e,i}(x, \gamma)$ is equal to one, which gives

$$\sigma_i^2(\gamma) = \frac{2}{\pi}(x_i(\gamma) - x_j(\gamma))^2 \ .$$

Finally, for a single sample, we obtain

$$f_{e,i}(x, \gamma) = \underbrace{\frac{1}{2}\frac{w_i}{|x_i(\gamma) - x_j(\gamma)|}}_{\text{density value at sample}} \underbrace{\exp\left(-\frac{\pi}{4}\frac{(x - x_i(\gamma))^2}{(x_i(\gamma) - x_j(\gamma))^2}\right)}_{\text{interpolation between samples}} \ . \tag{17}$$

For all samples, we have

$$f_e(x, \gamma) = k(\gamma) \sum_{i=1}^{L} f_{e,i}(x, \gamma) \ ,$$

where $k(\gamma)$ is the factor that takes care of the fact that we calculate an unnormalized solution of the DODE (11).

6.1. Derivation of DODE for Sample Locations

We will now derive a distributed ordinary differential equation (DODE) that describes the evolution of the sample locations for the measurement update step along the artificial time $\gamma \in [0, 1]$. The derivative of $f_e(x, \gamma)$ with respect to γ is

$$\begin{aligned}
\dot{f}_e(x, \gamma) &= \dot{k}(\gamma) \sum_{i=1}^{L} f_{e,i}(x, \gamma) + k(\gamma) \sum_{i=1}^{L} \dot{f}_{e,i}(x, \gamma) \\
&= \frac{f_e(x, \gamma)}{k(\gamma)}\dot{k}(\gamma) + k(\gamma) \sum_{i=1}^{L} \dot{f}_{e,i}(x, \gamma) \ .
\end{aligned} \tag{18}$$

The derivatives of $f_{e,i}(x, \gamma)$ in (17) with respect to γ are given by

$$\dot{f}_{e,i}(x, \gamma) = \frac{\partial f_{e,i}(x, \gamma)}{\partial \gamma} = \underbrace{\begin{bmatrix} \frac{\pi x^2 - 2x_i^2(\gamma) - x_j(\gamma)\left(\pi x + 2x_j(\gamma)\right) + x_i(\gamma)\left((4+\pi)x_j(\gamma) - \pi x\right)}{2\left(x_i(\gamma) - x_j(\gamma)\right)^3} \\ -\frac{\pi x^2 - 2\pi x x_i(\gamma) + \pi x_i^2(\gamma) - 2\left(x_i(\gamma) - x_j(\gamma)\right)^2}{2\left(x_i(\gamma) - x_j(\gamma)\right)^3} \end{bmatrix}^\top}_{\underline{b}_i^\top(x, \gamma)} f_{e,i}(x, \gamma) \begin{bmatrix} \dot{x}_i(\gamma) \\ \dot{x}_j(\gamma) \end{bmatrix}$$

15

for $i = 1, \ldots, L$. We note that $\dot{f}_{e,i}(x,\gamma)$ only depends on the location $x_i(\gamma)$ with index i (and its derivative $\dot{x}_i(\gamma)$)) and on the location $x_j(\gamma)$ with index j, the nearest neighbor, (and its derivative $\dot{x}_j(\gamma)$). The components of the vector $\underline{b}_i(x,\gamma)$ will be named as follows for the subsequent derivations

$$\underline{b}_i^\top (x,\gamma) = \begin{bmatrix} R_i(x,\gamma) & S_i(x,\gamma) \end{bmatrix} \ .$$

It can be rewritten as

$$\underline{b}_i^\top (x,\gamma) = \begin{bmatrix} 1 & x & x^2 \end{bmatrix} \mathbf{A}_i(\gamma)\, f_{e,i}(x,\gamma)$$

with

$$\mathbf{A}_i(\gamma) = \frac{1}{2\left(x_i(\gamma) - x_j(\gamma)\right)^3} \begin{bmatrix} (4+\pi)\,x_i(\gamma)\,x_j(\gamma) - 2\left(x_i^2(\gamma) + x_j^2(\gamma)\right) & 2\left(x_i(\gamma) - x_j(\gamma)\right)^2 - \pi\,x_i^2(\gamma) \\ -\pi\left(x_i(\gamma) + x_j(\gamma)\right) & 2\,\pi\,x_i(\gamma) \\ \pi & -\pi \end{bmatrix} \ .$$

The derivative of $\dot{f}_e(x,\gamma)$ in (18) can now be expressed as

$$\dot{f}_e(x,\gamma) = \frac{f_e(x,\gamma)}{k(\gamma)}\dot{k}(\gamma) + k(\gamma) \sum_{i=1}^{L} \begin{bmatrix} R_i(x,\gamma) & S_i(x,\gamma) \end{bmatrix} \begin{bmatrix} \dot{x}_i(\gamma) \\ \dot{x}_j(\gamma) \end{bmatrix} \ .$$

which is equivalent to

$$\dot{f}_e(x,\gamma) = \frac{f_e(x,\gamma)}{k(\gamma)}\dot{k}(\gamma) + k(\gamma) \sum_{i=1}^{L} [0,0,\ldots,0,\underbrace{R_i(x,\gamma)}_{\text{position } i},0,\ldots,0,0] \begin{bmatrix} \dot{x}_1(\gamma) \\ \dot{x}_2(\gamma) \\ \vdots \\ \dot{x}_L(\gamma) \end{bmatrix}$$
$$+ k(\gamma) \sum_{i=1}^{L} [0,\ldots,0,0,\underbrace{S_i(x,\gamma)}_{\text{position } j},0,0,\ldots,0] \begin{bmatrix} \dot{x}_1(\gamma) \\ \dot{x}_2(\gamma) \\ \vdots \\ \dot{x}_L(\gamma) \end{bmatrix} \ .$$

The summations can be simplified to

$$\dot{f}_e(x,\gamma) = \frac{f_e(x,\gamma)}{k(\gamma)}\dot{k}(\gamma) + k(\gamma) \begin{bmatrix} R_1(x,\gamma), R_2(x,\gamma), \ldots, R_L(x,\gamma) \end{bmatrix} \begin{bmatrix} \dot{x}_1(\gamma) \\ \dot{x}_2(\gamma) \\ \vdots \\ \dot{x}_L(\gamma) \end{bmatrix}$$
$$+ k(\gamma) \begin{bmatrix} T_1(x,\gamma), T_2(x,\gamma), \ldots, T_L(x,\gamma) \end{bmatrix} \begin{bmatrix} \dot{x}_1(\gamma) \\ \dot{x}_2(\gamma) \\ \vdots \\ \dot{x}_L(\gamma) \end{bmatrix}$$

with

$$T_j(\gamma) = \sum_{\substack{i=1 \\ j=\text{NN}(i)}}^{L} S_i(\gamma)$$

for $j = 1, \ldots, L$, which leads to

$$\dot{f}_e(x,\gamma) = \mathbf{p}^\top(x,\gamma)\,\underline{\dot{\eta}}(\gamma) \tag{19}$$

16

with

$$
\mathbf{p}(x,\gamma) = \begin{bmatrix} \frac{f_e(x,\gamma)}{k(\gamma)} \\ k(\gamma)\big(R_1(x,\gamma)+T_1(x,\gamma)\big) \\ k(\gamma)\big(R_2(x,\gamma)+T_2(x,\gamma)\big) \\ \vdots \\ k(\gamma)\big(R_L(x,\gamma)+T_L(x,\gamma)\big) \end{bmatrix} \quad , \quad \underline{\dot{\eta}}(\gamma) = \begin{bmatrix} \dot{k}(\gamma) \\ \dot{x}_1(\gamma) \\ \dot{x}_2(\gamma) \\ \vdots \\ \dot{x}_L(\gamma) \end{bmatrix} .
$$

The DODE for the sample locations is now obtained by equating (19) with the left-hand side given in Sec. 3.

6.2. Solution of DODE for Sample Locations based on Collocation Points

Solving this DODE is again performed by solving over a discrete set of points. Here, we focus on employing collocation points for an average solution over the state domain. This leads to a system of coupled ordinary differential equations for the desired sample locations. The collocation points are selected as follows. For each sample point $x_{e,i}(\gamma)$, a Dirac mixture approximation of the corresponding interpolation density $f_{e,i}(x,\underline{\eta}(\gamma))$ is performed. These points x_n for $n=1,\dots,C$ are plugged into (19), which gives

$$
\underline{q}\big(\underline{\eta}(\gamma)\big) = \mathbf{P}\big(\underline{\eta}(\gamma)\big) \cdot \underline{\dot{\eta}}(\gamma) \ ,
$$

with

$$
\mathbf{P}\big(\underline{\eta}(\gamma)\big) = \begin{bmatrix} \mathbf{p}^\top(x_1,\gamma) \\ \mathbf{p}^\top(x_2,\gamma) \\ \vdots \\ \mathbf{p}^\top(x_C,\gamma) \end{bmatrix} .
$$

7. Conclusions

The proposed new density approximation method generates a set of ordinary differential equations for the evolution of a parameter vector over an artificial time $\gamma \in [0,1]$ that starts with an approximation for the prior density for $\gamma = 0$ and results in the desired approximation of the posterior for $\gamma = 1$. It provides some unique features compared with the current state of the art that make it well suited for practical applications:

- The derivations are easy to understand.

- Implementation is straightforward and can be done based on standard ODE solvers.

- It is applicable to a wide variety of density approximation problems, where the approximated density does not even have to be known explicitly. This includes approximating the posterior densities in the Bayesian filter step in stochastic nonlinear dynamic systems.

- Solving for the desired posterior density parameters such as particle locations is inherently fast as neither an optimization nor solving a partial differential equation is required.

Future work includes higher dimensions, varying the length of the parameter vector during the progression, and real-time implementations:

- The derivations in this manuscript are limited on the scalar case for the sake of simplicity. The next step is the generalization to a higher number of dimensions.

17

- The generated flow modifies the values of a parameter vector of fixed length in such a way that the posterior density is well approximated. However, for both parametric and nonparametric representations, an adaptation of the number of parameters might be required. For a particle representation, this means adding or removing particles depending on the local approximation quality.

- So far, the algorithms are implemented in Matlab. For real-time operation, we envision to auto-generate C++ code that can be compiled for different platforms.

References

[1] F. Daum and J. Huang, "Particle Degeneracy: Root Cause and Solution," in *Proc. SPIE, Signal Processing, Sensor Fusion, and Target Recognition XX*, vol. 8050, 2011.

[2] N. Oudjane and C. Musso, "Progressive Correction for Regularized Particle Filters," in *Proceedings of the Third International Conference on Information Fusion (Fusion 2000)*, vol. 2, Jul. 2000, pp. 10 – 17.

[3] R. M. Neal, "Annealed Importance Sampling," *Statistics and Computing*, vol. 11, no. 2, pp. 125–139, Apr. 2001.

[4] P. Del Moral, A. Doucet, and A. Jasra, "Sequential Monte Carlo Samplers," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 3, pp. 411–436, Jun. 2006.

[5] J. Gall, J. Potthoff, C. Schnörr, B. Rosenhahn, and H.-P. Seidel, "Interacting and Annealing Particle Filters: Mathematics and a Recipe for Applications," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 1, pp. 1–18, May 2007.

[6] F. Daum and J. Huang, "Nonlinear Filters with Log-Homotopy," in *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 6699, 2007.

[7] ——, "Nonlinear Filters with Particle Flow," in *Proc. SPIE, Signal and Data Processing of Small Targets*, vol. 7445, 2009.

[8] ——, "Nonlinear Filters with Particle Flow Induced by Log-Homotopy," in *Proc. SPIE, Signal Processing, Sensor Fusion, and Target Recognition XVIII*, vol. 7336, 2009.

[9] F. Daum, J. Huang, and A. Noushin, "Exact Particle Flow for Nonlinear Filters," in *Proc. SPIE, Signal Processing, Sensor Fusion, and Target Recognition XIX*, vol. 7697, 2010.

[10] ——, "Numerical Experiments for Coulomb's Law Particle Flow for Nonlinear Filters," vol. 8137, 2011.

[11] F. Daum and J. Huang, "Particle Flow Inspired by Knothe-Rosenblatt Transport for Nonlinear Filters," in *Proc. SPIE, Signal Processing, Sensor Fusion, and Target Recognition XXII*, vol. 8745, 2013.

[12] M. Khan and M. Ulmke, "Non-Linear and Non-Gaussian State Estimation Using Log-Homotopy Based Particle Flow Filters," in *Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2014*, Oct. 2014, pp. 1–6.

[13] P. Bunch and S. Godsill, "Approximations of the Optimal Importance Density Using Gaussian Particle Flow Importance Sampling," *arXiv:1406.3183 [stat]*, Jun. 2014.

[14] T. Yang, P. G. Mehta, and S. P. Meyn, "Feedback Particle Filter," *arXiv:1302.6563*, Feb. 2013.

[15] K. Berntorp and P. Grover, "Feedback Particle Filter with Data-Driven Gain-Function Approximation," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2018.

[16] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[17] J. Steinbring and U. D. Hanebeck, "S2KF: The Smart Sampling Kalman Filter," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[18] ——, "LRKF Revisited: The Smart Sampling Kalman Filter (S2KF)," *Journal of Advances in Information Fusion*, vol. 9, no. 2, pp. 106–123, Dec. 2014.

[19] J. Steinbring, M. Pander, and U. D. Hanebeck, "The Smart Sampling Kalman Filter with Symmetric Samples," *Journal of Advances in Information Fusion*, vol. 11, no. 1, pp. 71–90, Jun. 2016.

[20] K. Berntorp, "Feedback Particle Filter: Application and Evaluation," in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Jul. 2015, pp. 1633–1640.

[21] U. D. Hanebeck, K. Briechle, and A. Rauh, "Progressive Bayes: A New Framework for Nonlinear State Estimation," in *Proceedings of SPIE, AeroSense Symposium*, vol. 5099, Orlando, Florida, USA, May 2003, pp. 256 – 267.

[22] U. D. Hanebeck, "Progressive Bayesian Estimation for Nonlinear Discrete-Time Systems: The Measurement Step," in *Proceedings of the 2003 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2003)*, Tokyo, Japan, Jul. 2003, pp. 173–178.

[23] U. D. Hanebeck and O. Feiermann, "Progressive Bayesian Estimation for Nonlinear Discrete-Time Systems: The Filter Step for Scalar Measurements and Multidimensional States," in *Proceedings of the 2003 IEEE Conference on Decision and Control (CDC 2003)*, Maui, Hawaii, USA, Dec. 2003, pp. 5366–5371.

[24] J. Hagmar, M. Jirstrand, L. Svensson, and M. Morelande, "Optimal Parameterization of Posterior Densities Using Homotopy," in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*. IEEE, Jul. 2011, pp. 1–8.

[25] A. Rauh and U. D. Hanebeck, "Calculating Moments of Exponential Densities Using Differential Algebraic Equations," *IEEE Signal Processing Letters*, vol. 10, no. 5, pp. 144–147, May 2003.

[26] M. F. Huber and U. D. Hanebeck, "Progressive Gaussian Mixture Reduction," in *Proceedings of the 11th International Conference on Information Fusion (Fusion 2008)*, Cologne, Germany, Jul. 2008, pp. 1–8.

[27] U. D. Hanebeck and J. Steinbring, "Progressive Gaussian Filtering," *arXiv preprint: Systems and Control (cs.SY)*, Mar. 2012. [Online]. Available: http://arxiv.org/abs/1204.0133

[28] ——, "Progressive Gaussian Filtering Based on Dirac Mixture Approximations," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

[29] P. Ruoff, P. Krauthausen, and U. D. Hanebeck, "Progressive Correction for Deterministic Dirac Mixture Approximations," in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, Illinois, USA, Jul. 2011.

[30] U. D. Hanebeck and M. Pander, "Progressive Bayesian Estimation with Deterministic Particles," in *Proceedings of the 19th International Conference on Information Fusion (Fusion 2016)*, Heidelberg, Germany, Jul. 2016.

[31] U. D. Hanebeck, "Optimal Reduction of Multivariate Dirac Mixture Densities," *at - Automatisierungstechnik, Oldenbourg Verlag*, vol. 63, no. 4, pp. 265–278, Apr. 2015. [Online]. Available: http://dx.doi.org/10.1515/auto-2015-0005

[32] ——, "PGF 42: Progressive Gaussian Filtering with a Twist," in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[33] J. Steinbring and U. D. Hanebeck, "Progressive Gaussian Filtering Using Explicit Likelihoods," in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[34] J. Steinbring, A. Zea, and U. D. Hanebeck, "Semi-Analytic Progressive Gaussian Filtering," in *Proceedings of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, Baden-Baden, Germany, Sep. 2016.

[35] J. Steinbring and U. D. Hanebeck, "GPU-Accelerated Progressive Gaussian Filtering with Applications to Extended Object Tracking," in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, Jul. 2015.