GRID METHODS FOR BAYES-OPTIMAL CONTINUOUS-DISCRETE FILTERING AND UTILIZING A FUNCTIONAL TENSOR TRAIN REPRESENTATION

Colin Fox[a], Sergey Dolgov[b], Malcolm E. K. Morrison[c], Timothy C. A. Molteno[d]

[a]*Electronics Group, University of Otago, Dunedin, New Zealand colin.fox@otago.ac.nz*
[b]*Department of Mathematical Sciences, University of Bath, United Kingdom s.dolgov@bath.ac.uk*
[c]*Electronics Group, University of Otago, Dunedin, New Zealand malcolmerikking@gmail.com*
[d]*Electronics Group, University of Otago, Dunedin, New Zealand tim@physics.otago.ac.nz*

Abstract

Optimal continuous-discrete filtering for a nonlinear system requires evolving the forward Kolmogorov equation, that is a Fokker–Planck equation, in alternation with Bayes' conditional updating. We present two numerical grid-methods that represent density functions on a mesh, or grid. For low-dimensional, smooth systems the finite-volume method is an effective solver that gives estimates that converge to the optimal continuous-time values. We give numerical examples to show that this finite-volume filter is able to handle multi-modal filtering distributions that result from rank-deficient observations, and that Bayes-optimal parameter estimation may be performed within the filtering process. The naïve discretization of density functions used in the finite-volume filter leads to exponential increase of computational cost and storage with increasing dimension, that makes the finite-volume filter unfeasible for higher dimensional problems. We circumvent this 'curse of dimensionality' by using a tensor train representation (or approximation) of density functions and employ an efficient implicit PDE solver that operates on the tensor train representation. We present numerical examples of tracking n weakly coupled pendulums in continuous time to demonstrate filtering with complex density functions in up to 80 dimensions.

I. Introduction: Filtering

The notion of a *filter* originated in signal processing, with electronic circuits aimed at modifying frequency content. For early developments in radar, filters were developed that could extract 'signals' from 'noise', with two examples being the Wiener filter and the Middleton-North (-Van Vleck) matched filter. Each of these filters is also linear, time-invariant, and causal, so can be implemented using (analog) electronic circuits. Antecedents of the matched filters are used in present-day gravitational wave detectors.

For applications in radar, the notion of filtering was extended to the task of real-time *tracking* of a target from noisy measurements, which requires determining the time-varying vector of kinematic variables of the target, typically location and velocity. The most widely used example is the Kalman filter that is now ubiquitous in navigation. These filters maintain vector quantities, and matrices, so are usually implemented on a computer, though one might, conceivably build an analog circuit for a fixed application.

The filters presented in this paper perform the task of tracking a target, where kinematics of the target is provided through a dynamical system equation. We consider both deterministic and stochastic equations of motion; the stochastic term can be used to model either unknown 'microscopic' forces acting on the 'macroscopic' target, or for modelling error introduced by reduced-order or computational models. The simultaneous tracking of multiple objects, termed multiple object/target tracking, is a modern concern and is a potential application for the filters we present here; see section II.

In a statistical analysis, that also determines uncertainties in estimates, it is necessary to determine the time-dependent probability distribution over state variables, rather than just best estimates. We take a Bayesian statistical approach, which is to say that we model all noise or uncertainties as random variables with probability distributions. The mathematical requirements for Bayesian filtering, or sequential inference, have been known for more than half a century [1], however computational challenges have been prohibitive for general applications, particularly for multi-dimensional state spaces. Methods that implement the filtering equations exactly, or effectively

exactly, are termed 'optimal', other wise a method is termed 'sub-optimal' [2]. This paper reports some recent developments (by the authors) in Bayes-optimal filtering with possibly nonlinear dynamics and non-Gaussian random processes, and when the state variable is high dimensional.

The two filters presented here are 'grid methods' in which probability density functions (PDFs) are represented on a grid, or mesh, in state space. Grid methods were recognized early on as a way of implementing optimal filters [3] though the exponential explosion of storage and compute costs with increasing dimension incurred by a naïve grid representation meant that grid methods have been considered unfeasible [4]. Nevertheless, a few 'hard-boiled' engineers who are at the cutting edge of practical radar systems have stated that forming and solving the PDE version of the prediction step, as we do here, is the most promising route forward for practical and accurate tracking [5], [6], [7], [8].

For low-dimensional problems we solve that PDE using the finite-volume method; see Sec. II. For higher-dimensional applications we introduce a representation (or approximation) of PDFs in the tensor train (TT) format that allows low-rank approximations with storage that scales *linearly* with problem size, and dimension; see Sec. III. The attraction of approximating the PDF in TT format is that the computational cost of the construction, the storage requirements, and the operations required for filtering all scale *polynomial* with system dimension; see Sec. IV. This is a remarkable feature of the TT representation, and is why the recent introduction of low-rank hierarchical tensor methods, such as TT [9], [10], [11], [12], is a significant development in scientific computing for multi-dimensional problems. Tensor decomposition methods have also been applied to continuous-continuous filtering in [13]; we present a comparative discussion of that paper and conclusions in Sec. V.

A. Dynamical System and Observation Process

Consider a dynamical system that evolves according to the stochastic Langevin differential equation, written in Itô form (see, e.g., [13, Eq. (1)] or [1, Eq. (4.4)]),

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x})\mathrm{d}t + \mathrm{d}\boldsymbol{v} \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^d$ is the state vector at time t, $\boldsymbol{f}$ is a known velocity field, and $\mathrm{d}\boldsymbol{v}$ denotes an uncorrelated or 'white' noise increment defined in terms of the vector Wiener process $\boldsymbol{v}(t)$, i.e. $\boldsymbol{v}$ is a stochastic process with increments over a time interval Δt that are normally distributed with zero mean and covariance $\Delta t\,\mathrm{Cov}(\boldsymbol{v})$, where the matrix $\mathrm{Cov}(\boldsymbol{v})$ gives the covariance structure between components for unit time increment.

At increasing discrete times t_k, $k = 1, 2, 3, \ldots$, the system is observed, returning measurement $\boldsymbol{z}_k$ that provides noisy and incomplete information about $\boldsymbol{x}_k = \boldsymbol{x}(t_k)$. Assume that we know the conditional distribution over observed value $\boldsymbol{z}_k$, given the state $\boldsymbol{x}_k$,

$$\rho_{\text{obs}}\left(\boldsymbol{z}_k|\boldsymbol{x}_k\right).$$

That is, the measurement process and the statistics of the measurement noise are known.

Let $Z_t = \{\boldsymbol{z}_k : t_k \leq t\}$ denote the set of observations up to time t, and let (the random variable) $\boldsymbol{x}_t$ denote the unknown state at time t. The formal Bayesian solution corresponds to determining the time-varying sequence of 'filtering' distributions

$$\rho_{\text{filt}}\left(\boldsymbol{x}_t|Z_t\right) \tag{2}$$

over the state at time t conditioned on all measurements up to time t.

B. "Recursive" Bayesian filtering

Sequential Bayesian inference iterates two steps to generate the filtering distributions (2) [1].

a) Prediction: Between measurements times t_k and t_{k+1}, Z_t is constant and the continuous-time evolution of the filtering distribution may be derived from the (forward) Chapman-Kolmogorov equation

$$\begin{aligned}\rho_{\text{filt}}\left(\boldsymbol{x}_{t+\Delta t}|Z_{t+\Delta t}\right) &= \rho_{\text{filt}}\left(\boldsymbol{x}_{t+\Delta t}|Z_t\right) \\ &= \int \rho_{\text{evol}}(\boldsymbol{x}_{t+\Delta t}|\boldsymbol{x}_t)\rho_{\text{filt}}\left(\boldsymbol{x}_t|Z_t\right)\,\mathrm{d}\boldsymbol{x}_t,\end{aligned} \tag{3}$$

where $\rho_{\text{evol}}(\boldsymbol{x}_{t+\Delta t}|\boldsymbol{x}_t)$ is the probability (density) of finding the system is state $\boldsymbol{x}_{t+\Delta t}$ at time $t+\Delta t$ given that it started (certainly) in state $\boldsymbol{x}_t$ at time t and evolved according to (1). For deterministic systems, i.e., when $\boldsymbol{v} \equiv 0$, we may write $\rho_{\text{evol}}(\boldsymbol{x}_{t+\Delta t}|\boldsymbol{x}_t) = \delta(\boldsymbol{x}_{t+\Delta t} - \boldsymbol{x}(\Delta t; \boldsymbol{x}_t))$ where the notation $\boldsymbol{x}(t;\boldsymbol{x}_0)$ denotes the deterministic solution at time $t > 0$ when the dynamical system (1) is solved with initial state is $\boldsymbol{x}(0) = \boldsymbol{x}_0$ at $t = 0$. Equation (3) defines a linear operator on the space of probability distributions, that we write

$$T_{\Delta t} : \rho_{\text{filt}}\left(\boldsymbol{x}_t|Z_t\right) \mapsto \rho_{\text{filt}}\left(\boldsymbol{x}_{t+\Delta t}|Z_t\right). \tag{4}$$

$T_{\Delta t}$ is called the transfer operator[1], associated with (1), for time increment Δt.

b) Update: At measurement times t_k, Z_t changes, from Z_{k-1} to Z_k, and the filtering distribution changes, typically discontinuously, as

$$\rho_{\text{filt}}\left(\boldsymbol{x}_k|Z_k\right) = \frac{\rho_{\text{obs}}\left(\boldsymbol{z}_k|\boldsymbol{x}_k\right)\rho_{\text{filt}}\left(\boldsymbol{x}_k|Z_{k-1}\right)}{\rho_{\text{norm}}\left(\boldsymbol{z}_k|Z_{k-1}\right)}, \tag{5}$$

which is simply Bayes' rule written at observation time t_k. The normalizing constant is the marginal density $\rho_{\text{norm}}\left(\boldsymbol{z}_k|Z_{k-1}\right) = \int \rho_{\text{obs}}\left(\boldsymbol{z}_k|\boldsymbol{x}_k\right)\rho_{\text{filt}}\left(\boldsymbol{x}_k|Z_{k-1}\right)\,\mathrm{d}\boldsymbol{x}_k$ evaluated at the new observation $\boldsymbol{z}_k$. We have written $\boldsymbol{x}_k = \boldsymbol{x}_{t_k}$ and $Z_k = Z_{t_k}$, and used conditional independence of $\boldsymbol{z}_k$ and Z_{k-1} given $\boldsymbol{x}_k$.

Practical filtering in a particular application also requires computing the posterior expectation of some relevant function of the state $g(\boldsymbol{x})$ at time t as

$$\mathrm{E}\left[g|Z_t\right] = \int g(\boldsymbol{x}_t)\rho_{\text{filt}}\left(\boldsymbol{x}_t|Z_t\right)\,\mathrm{d}\boldsymbol{x}_t. \tag{6}$$

The set of statistics that usefully summarize the posterior filtering distributions is application specific. In this paper we demonstrate on the computation of full filtering densities and do not further consider summary statistics or estimates.

C. PDE for PDFs

The transfer operator (4), that evolves the filtering density forward in time, may be written as the solution of an initial value problem (IVP) in the partial differential equation (PDE) for the probability density function (PDF). In this section, and in Sec. II-D, only, we will denote a general density that depends on state $\boldsymbol{x}$ and time t by $\rho(x;t)$ in keeping with notation for PDEs.

By writing the movement of the density over very short, or infinitesimal, times we can derive the transfer operator for infinitesimal times in the form of a partial differential equation, as follows. We first consider the right-hand side of (1) with just the flow term $\boldsymbol{f}(\boldsymbol{x})$. The density $\rho(x;t)$ and velocity field $\boldsymbol{f}(\boldsymbol{x})$ implies a flux of probability equal to $\rho(\boldsymbol{x};t)\boldsymbol{f}(\boldsymbol{x})$. Fig. 1 shows a schematic of the PDF and probability flux in region $(\boldsymbol{x}, \boldsymbol{x}+\mathrm{d}\boldsymbol{x})$, and for the time interval $(t, t+\mathrm{d}t)$. Equating the rate of change in the PDF with the rate at which probability mass enters the region, and taking $|\mathrm{d}\boldsymbol{x}|, \mathrm{d}t \to 0$, gives the continuity equation $\frac{\partial}{\partial t}\rho = -\nabla\cdot(\rho\boldsymbol{f})$. The effect of the random forcing is to blur-out the density; the integral of the random forcing over period $\mathrm{d}t$ results in adding the

[1]The transfer operator for deterministic systems ($\boldsymbol{v} \equiv 0$) is specifically called the Frobenius–Perron operator, while for systems with stochastic perturbation the transfer operator on the measure is called the Foias operator [14].

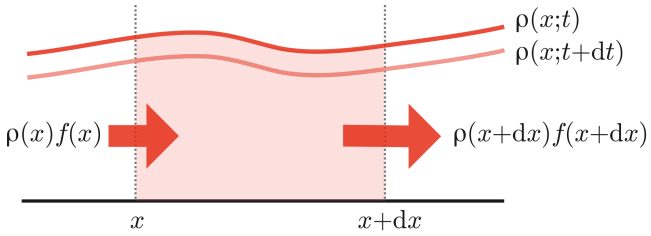

Fig. 1. A schematic of probability flux in region $(\boldsymbol{x}, \boldsymbol{x}+\mathrm{d}\boldsymbol{x})$, and for time $(t, t+\mathrm{d}t)$. The schematic shows greater flux exiting the region than entering, correspondingly the PDF at $t+\mathrm{d}t$ is decreased with respect to the PDF at t.

independent Wiener process increment $\mathrm{d}\boldsymbol{v} \sim \mathrm{MVN}(0, \mathrm{d}t\,\mathrm{Cov}(\boldsymbol{v}))$. Since addition of independent random variables corresponds to convolution of PDFs, the random forcing results in convolution with the PDF of $\mathrm{MVN}(0, \mathrm{d}t\,\mathrm{Cov}(\boldsymbol{v}))$, the latter being the fundamental solution of the diffusion operator $\frac{\partial}{\partial t} - \frac{1}{2}\sum_{ij} \frac{\partial^2}{\partial x_i \partial x_j}\mathrm{Cov}(\boldsymbol{v})$. Combining terms, using linearity of the derivative, gives the infinitesimal generator for the transfer operator as

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \boldsymbol{f}) + \frac{1}{2}\sum_{ij} \frac{\partial^2}{\partial x_i \partial x_j}\mathrm{Cov}(\boldsymbol{v})\rho. \tag{7}$$

In physics, this equation is called the Fokker–Planck equation associated with the dynamical system (1). The transfer operator $T_{\Delta t}$, for time interval Δt, may be simulated by solving the PDE (7) with initial condition $\rho(\boldsymbol{x}; 0) = \rho_{\text{filt}}(\boldsymbol{x}_t | Z_t)$ to evaluate $\rho(\boldsymbol{x}; \Delta t) = \rho_{\text{filt}}(\boldsymbol{x}_{t+\Delta t} | Z_t)$.

Note that the continuity equation for deterministic systems is a linear advection equation, while the additive stochastic forcing results in the linear advection-diffusion (Fokker–Planck) equation. This is significant for computation, as the PDE (7) over PDFs that needs to be solved is a deterministic linear equation, even if the underlying dynamics (1) is nonlinear and stochastic.

II. Finite Volume Filter : tracking a simple pendulum

We present filtering for the simple pendulum using a filter based on the finite-volume method of discretization and the (simplest-possible) explicit Euler integrator in time. This numerical method is often used for entry-level computation in fluid dynamics; see [15] for details. We refer to the resulting filter as the finite volume filter (FVF).

A. The simple pendulum

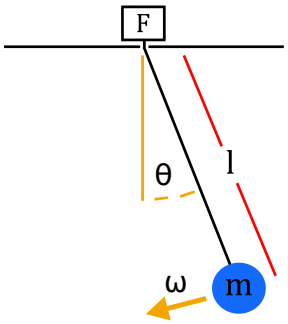

Fig. 2. A pendulum of length l with mass m, undergoing motion with angular displacement θ and angular velocity ω.

For this system the kinematic variables are the angular displacement (from the vertical downwards) θ and the angular velocity ω. The kinematic state $\boldsymbol{x} = (\theta, \omega)^T$ evolves according to

$$\frac{\mathrm{d}}{\mathrm{d}t}(\theta, \omega)^T = \left(\omega, -\frac{g}{l}\sin(\theta)\right)^T = \boldsymbol{f}(\boldsymbol{x})$$

where g is the acceleration due to gravity, l the length of the pendulum. This is a deterministic system, so the noise term in (1) is zero.

It is possible to perform parameter estimation by filtering, within the framework for sequential Bayesian inference described above. This is done by augmenting the state with the (unknown) parameter of interest, so that state estimation also returns the parameter. For example, we can perform filtering for both kinematic state and unknown length l of a simple pendulum, by augmenting the state vector to be $\boldsymbol{x} = (\theta, \omega, l)^T$, and so the velocity field is

$$\frac{\mathrm{d}}{\mathrm{d}t}(\theta, \omega, l)^T = \left(\omega, -\frac{g}{l}\sin(\theta), 0\right)^T = \boldsymbol{f}(\boldsymbol{x}). \tag{8}$$

The parameter l is modelled as having a fixed value, so has zero time derivative.

B. Tracking the pendulum

We present an example of tracking the pendulum using noisy measurements of the absolute value of angle, $|\theta|$. Interestingly, these observations lead to multi-modal filtering distributions.

We simulated the 'true' pendulum of length $l = 1$, and using $g = 1$, with initial condition $(\theta, \omega) = (0.5\pi, 0)$ for the time interval $[0, 6\pi]$ and then took noisy observations of $|\theta|$ at the 12 times $t_k = k\pi/2$, $k = 1, 2, \ldots, 12$, each observation drawing from $z_k \sim \mathrm{N}(|\theta(t_k)|, 0.1^2)$. Since the absolute value of angle is measured, measurements provide no information about the sign of the angular displacement θ, or the sign of angular velocity ω.

For filtering, we assumed an initial prior distribution that is a Gaussian in (θ, ω) with mean $(0, 0)$ and covariance $0.64\mathbf{I}$, where $\mathbf{I}$ is the 2×2 identity matrix, and uniform in l over the interval $[0.8, 1.2]$. The prior PDF over (θ, ω), given $l = 1$, is shown in the top-left panel in Fig. 3. As can be seen, truncation to the computational domain is negligible.

We solve the continuity equation using a finite volume method discretization in space and explicit Euler time step; some computational details are given in Sec. II.

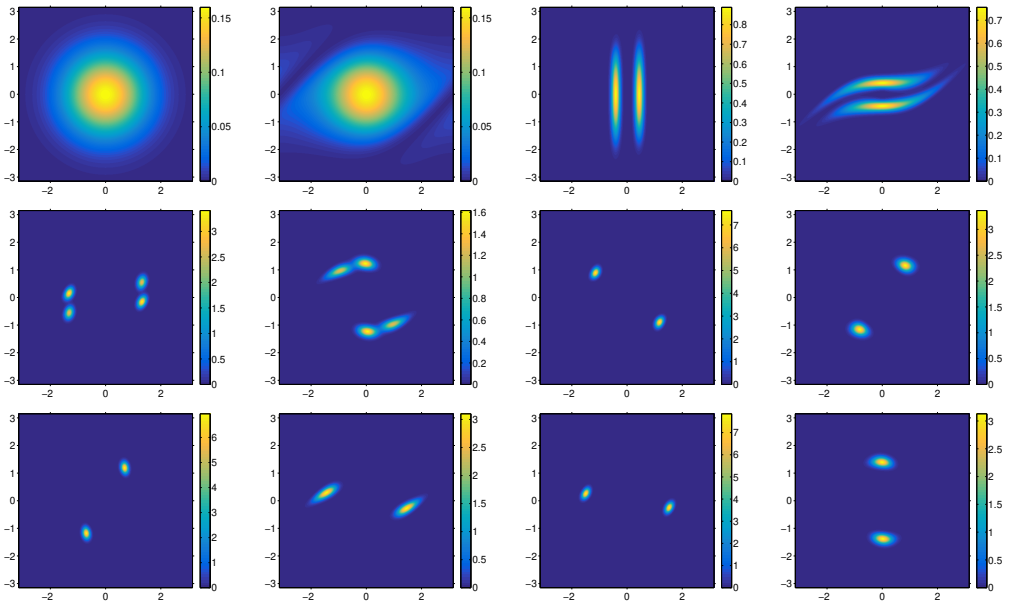

Fig. 3. Probability density functions produced within filtering for the state of the simple pendulum from measurements of absolute value of angle, $|\theta|$ at time intervals of $\pi/2$. Angle θ is shown horizontally, velocity ω vertically. Shown are (left to right, top to bottom) the prior distribution, then pairs of predictive and filtering distributions.

The remaining subfigures in Fig. 3 (left to right, top to bottom) show the predictive and filtering PDFs for the next six time steps of filtering, all conditioned on $l = 1$, as we would when l is known. The subfigure in position 1-2 (top row, second from left) shows the predictive PDF immediately

before the first measurement, so is the result of using the prior, subfigure 1-1, as initial condition in the PDE (7) then time stepping forward to t_1 to produce the predictive PDF in subfigure 1-2. The velocity field for the pendulum is basically a clockwise rotation of the phase plane, but since larger amplitudes have longer period those states rotate more slowly resulting in a 'cork-screw' pattern. At time t_1 a noisy measurement of absolute value of angle $|\theta|$ is made giving a likelihood that is two vertical stripes (plus or minus the measured angle) with width of the stripes set by the observation noise. Multiplying that likelihood with subfigure 1-2 and normalizing, i.e., performing the update step (5), produces subfigure 1-3, that is the filtering PDF at time t_1. Subfigure 1-4 is then the result of the next prediction step to time t_2, with the cork-screw pattern more evident. Subfigure 2-1 is then the result of the update step at time t_2, that again cuts two vertical slices of subfigure 1-4, resulting in the four modes that can be seen. The remaining subfigures continue this pattern of prediction then update.

Perhaps it is stating the obvious to note that this filter is able to handle multi-modal distributions, as required for multiple target tracking.

C. Determining Pendulum Length from Measured $|\theta|$

As mentioned above, it is also possible to determine the length of the pendulum by treating length l as an unknown parameter and augmenting the state as in (8).

For this parameter set, state space is the region $(\theta, \omega, l) \in [-\pi, \pi) \times \mathbb{R} \times \mathbb{R}^+$ (periodic in angle θ), hence $d = 3$. For computational purposes we restrict the computational domain to $(\theta, \omega, l) \in \Omega = [-\pi, \pi)^2 \times [0.8, 1.2]$ with periodic boundary conditions at $\theta = \pm\pi$, and homogeneous Neumann boundary conditions at $\omega = \pm\pi$ and at $l = 0.8, 1.2$. We discretized the PDFs on a square mesh in (θ, ω) having $n = 500$ cells in each of the θ and ω coordinates, and $n_l = 51$ equally spaced cells in the l coordinate [2].

Figure 4 shows the sequence of marginal distributions over length,

$$\rho_{\text{marg},l}(l_t|Z_t) = \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \rho_{\text{filt}}(\boldsymbol{x}_t|Z_t)\mathrm{d}\theta_t\,\mathrm{d}\omega_t.$$

Since this marginal distribution only changes at update steps, the figure shows the initial (uniform) distribution, and after each update step. Clearly, the measurements are informative about the true length, once the kinematic variables are located at time step 4, with the marginal distributions becoming increasingly peaked around the true value of $l = 1$.

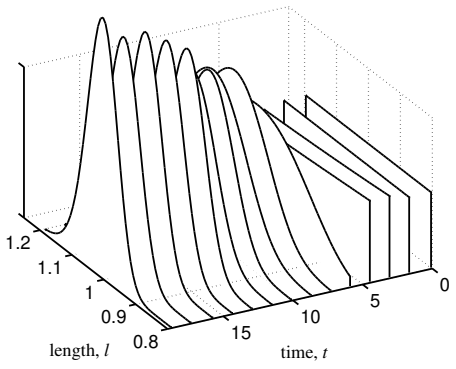

Fig. 4. Sequence of marginal distributions over l, for the initial PDF, and after each update due to a measurement. The initial distribution at $t = 0$ is uniform in l over the interval $[0.8, 1.2]$. As can be seen, the marginal PDF over l becomes increasingly peaked about the true value of $l = 1$ as time increases.

[2]Fig. 3 shows the conditional PDFs from this calculation for the cells with $l \approx 1$, so, strictly, $l \in [0.996, 1.004]$.

D. Finite Volume Solver

For completeness we present some very brief details of the finite volume solver that we used to perform the prediction step in the FVF results shown above. Further details and proofs of distributional convergence may be found in [16].

The finite volume method (FVM) discretizes the continuity equation in its integral form, for each 'cell' K in a mesh,

$$\frac{\partial}{\partial t}\int_K \rho \,\mathrm{d}\boldsymbol{x} + \oint_{\partial K} \rho(\boldsymbol{f}\cdot\hat{\boldsymbol{n}})\,\mathrm{d}S = 0.$$

Write $L \sim K$ if cells L and K share a common interface, denoted E_{KL}, and denote by $\hat{\boldsymbol{n}}_{KL}$ the unit normal on E_{KL} directed from K to L. Define the initial vector $\boldsymbol{P}^0 = \{\boldsymbol{P}^0_K\}$ of cell values by $\boldsymbol{P}^0_K = \frac{1}{|K|}\int_K \rho(\boldsymbol{x};0)\,\mathrm{d}\boldsymbol{x}$ then for $m = 0, 1, \cdots, r$ compute $\boldsymbol{P}^{m+1}$ as the Euler step

$$\boldsymbol{P}^{m+1} = (\boldsymbol{I} - \Delta t \boldsymbol{A})\,\boldsymbol{P}^m, \tag{9}$$

where $\boldsymbol{I}$ is the identity matrix and $\boldsymbol{A}$ is a sparse matrix defined component-wise by the first-order *upwinding* scheme [15]

$$\boldsymbol{A}_{KK} = \frac{1}{|K|}\sum_{L\sim K}\max\left(0, f_{KL}\right),$$

$$\boldsymbol{A}_{KL} = \begin{cases} \dfrac{1}{|K|}\min\left(0, f_{KL}\right), & \text{if } K\sim L,\\ 0, & \text{otherwise,}\end{cases}$$

where

$$f_{KL} = \int_{E_{KL}} \boldsymbol{f}\cdot\hat{\boldsymbol{n}}_{KL}\,\mathrm{d}S$$

is the normal velocity on E_{KL}.

Since $f_{KL} = -f_{LK}$, the FVM conserves probability at each step, i.e., $\sum_K |K|\boldsymbol{P}^{m+1}_K = \sum_K |K|\boldsymbol{P}^m_K$. The FVM also preserves positivity of the PDF when the time step Δt is small enough that the matrix $\boldsymbol{I} - \Delta t\boldsymbol{A}$ has all non-negative entries. This constraint may be written as the Courant-Friedrichs-Lewy (CFL) type condition

$$\Delta t \le \frac{1}{\max_K \boldsymbol{A}_{KK}},$$

where $\max_K$ denotes the maximum over cells K. With this condition, the FVM both conserves probability and is positivity preserving, hence is a (discrete) *Markov operator* [14]. This FVM achieves first-order convergence in time [16], which is best possible for explicit integrators such as (9) that preserve positivity, since such methods are necessarily limited to first-order convergence; see [17] and references therein.

III. PDFs in TT format

The results presented for the FVF, above, show that grid method filters are feasible, using standard PDE solvers to perform the prediction step, and can accommodate multi-modal PDFs and nonlinear dynamics, at least in low-dimensional settings. However, the FVF does not scale to higher dimensions.

The recent advent of tensor train (TT) representations, amongst other hierarchical matrix representations, allows us to circumvent this so-called 'curse of dimensionality', to give a Bayes-optimal sequential filter with cost that scales nearly linearly with dimension [18]. The use of an implicit time integrator – the tAMEn algorithm [19] – also allows higher than first-order convergence in time. In this section we give a gentle introduction to the representation of PDFs in TT format, and references to technical details for the interested reader.

A. Low-rank tensor decomposition

We first start with the low-rank representation of tensors, as multi-dimensional arrays are called in numerical linear algebra, and introduce the TT representation that we will use.

Consider a PDF $\rho(\boldsymbol{x})$ defined on a d-dimensional space that we wish to represent in a computer. We first consider the tensor that results from discretizing the PDF $\rho(x_1, \ldots, x_d)$ by collocation on a tensor product of univariate grids. Let $x_k^{i_k} \in \mathbb{R}$, with $i_k = 1, \ldots, n_k$ and $x_k^1 < \cdots < x_k^{n_k}$, define independent univariate grids in each variable, and let $\hat{\rho}(i_1, i_2, \ldots, i_d) = \rho(x_1^{i_1}, x_2^{i_2}, \ldots, x_d^{i_d})$.

This d-dimensional *tensor* $\hat{\rho}$ is depicted in Fig. 5. Simply storing all components of $\hat{\rho}$ costs n^d,

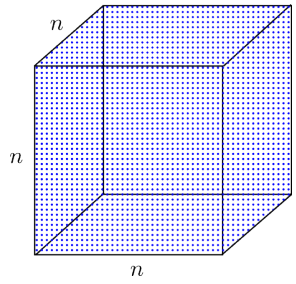

Fig. 5. The tensor $\hat{\rho}$ having n components in each of d dimensions.

which grows exponentially in d so quickly becomes too huge to be feasible.

However, if ρ is *smooth* then $\hat{\rho}$ is approximately low rank, and we can approximate $\hat{\rho}$ by a low-rank decomposition. This is the essential step in the reduction of storage and computational costs; we will utilize the regularity of 'typical' functions encountered in practical applications to make the computations feasible.

B. Singular-value decomposition (2-dim)

It is instructive to first consider the well-known low rank approximation used for matrices, i.e., when $d = 2$, based on the singular value decomposition (SVD).

The SVD of any matrix A is the representation

$$A = U\Sigma V^\top,$$

where U, V are orthonormal matrices, and Σ is a diagonal matrix of nonnegative singular values. The optimal rank r approximation to A (in the Frobenius norm) is given by the truncated SVD

$$\tilde{A} = U_r \Sigma_r V_r^\top$$

where Σ_r is a diagonal matrix containing the r largest singular values of A, and U_r, V_r contain the associated rows of U, V, respectively. This optimality result is called the Eckart-Young theorem [20]. The approximation is depicted in Fig. 6 for the case that A is $n \times n$, so U_r is $n \times r$ (tall and thin)

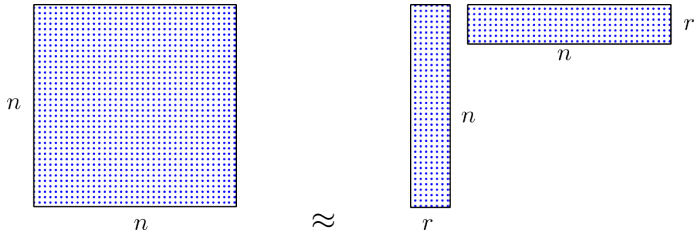

Fig. 6. The truncated SVD: $A \approx \left(U_r \Sigma_r^{1/2}\right)\left(\Sigma_r^{1/2} V_r^\top\right)$.

while $V_r^\top$ is $r \times n$ (short and wide); the diagonal matrix Σ_r has been absorbed into the two factors.

The truncated SVD has a few relevant pros and cons: a pro is that storage is $2nr \ll n^2$ when $r \ll n$ so storage is significantly reduced, while a con is that the full matrix needs to be in memory to compute the SVD. The last point is a show-stopper, because we have already seen that simply storing the full tensor $\hat{\rho}$ is not possible for large dimension d.

C. Skeleton decomposition (2-dim)

A more useful 'cross approximation' (the truncated SVD is also a cross approximation) may be derived from the skeleton decomposition [21], [9]. An equality from linear algebra, that is apparently much less well known than the SVD, is that any rank r matrix A can be written

$$A = A(:, \mathcal{J})A(\mathcal{I}, \mathcal{J})^{-1}A(\mathcal{I}, :)$$

where $\mathcal{I}$ and $\mathcal{J}$ are size r indices such that $A(\mathcal{I}, \mathcal{J})$ nonsingular. That is, it is always possible to find r columns of A to form the $n \times r$ matrix $A(:, \mathcal{J})$, and r rows of A to form the $r \times n$ matrix $A(\mathcal{I}, :)$ to give the skeleton decomposition above, where $A(\mathcal{I}, \mathcal{J})$ is the intersection of the rows and columns. Indeed, *any* r rows and columns for which the intersection $A(\mathcal{I}, \mathcal{J})$ is non-singular are admissible. The most stable, in some sense, is when the intersection matrix $A(\mathcal{I}, \mathcal{J})$ has the largest volume, that is the modulus of the determinant.

This motivates the rank r approximation depicted in Fig. 7 where r rows and columns are

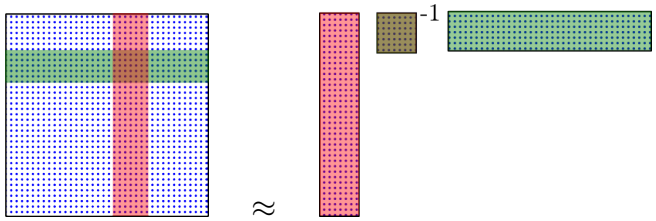

Fig. 7. Skeleton decomposition approximation: $A \approx A(:, \mathcal{J})A(\mathcal{I}, \mathcal{J})^{-1}A(\mathcal{I}, :)$.

chosen so that the intersection matrix $A(\mathcal{I}, \mathcal{J})$ has the largest volume amongst choices of rows and columns. See [21], [9] for details and practical algorithms.

This approximation has all the pros required for a practical algorithm that scales to higher dimensions. If $r \ll n$, this decomposition requires computing only $(2n - r)r \ll n^2$ elements of the original matrix, and it is not necessary to have the full matrix stored in memory (just the ability to compute elements).

D. Tensor Train (TT) decomposition

A suitable generalization to many variables is the tensor train (TT) decomposition [9], [11] that separates indices i_k by a product approximation of the form

$$\hat{\rho}(i_1, i_2, \ldots, i_d) = \sum_{\alpha_0, \ldots, \alpha_d = 1}^{r_0, \ldots, r_d} \hat{\rho}^{(1)}_{\alpha_0, \alpha_1}(i_1)\hat{\rho}^{(2)}_{\alpha_1, \alpha_2}(i_2) \cdots \hat{\rho}^{(d)}_{\alpha_{d-1}, \alpha_d}(i_d). \tag{10}$$

Here, $\hat{\rho}^{(k)}$, $k = 1, \ldots, d$, are called *TT cores*, and the summation ranges $r_1, \ldots, r_d$ are called *TT ranks*. We assume that all internal ranks $r_1, \ldots, r_{d-1}$ can be bounded by a moderate constant $r_k \leq r \ll n^d$, and for simplicity of exposition we will describe all these ranks as equal to r. Note that $r_0 = r_d = 1$ as $\hat{\rho}(i_1, \ldots, i_d)$ is scalar valued.

This decomposition is depicted in Fig. 8. The first and last cores are $n \times r$ and $r \times n$, just as in the skeleton decomposition (or the SVD). However, internal cores are 3-dimensional with size $r \times r \times n$.

The storage for this construction is $\mathcal{O}(dnr^2)$ which is *linear* in dimension d and problem size n. A practical algorithm for constructing this approximation is the TT-Cross [9], with cost that scales linearly with dimension d.

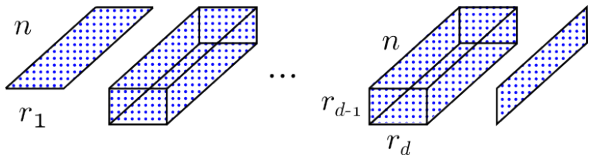

Fig. 8. Tensor train decomposition of a d-dimensional array, or tensor.

E. Interpolated Tensor Train representation of PDFs

Each PDF $\rho(\boldsymbol{x})$ is then approximated by interpolating the discrete TT decomposition (10) of expansion coefficients [18] $\hat{\rho}(i_1, \dots, i_d)$. For example, the linear interpolation of each TT core on the corresponding univariate grid gives

$$\rho^{(k)}_{\alpha_{k-1},\alpha_k}(x_k) = \frac{x_k - x_k^{i_k}}{x_k^{i_k+1} - x_k^{i_k}} \cdot \hat{\rho}^{(k)}_{\alpha_{k-1},\alpha_k}(i_k+1) + \frac{x_k^{i_k+1} - x_k}{x_k^{i_k+1} - x_k^{i_k}} \cdot \hat{\rho}^{(k)}_{\alpha_{k-1},\alpha_k}(i_k), \;\; x_k^{i_k} \le x_k \le x_k^{i_k+1},$$

which induces the multilinear approximation $\rho(\mathbf{x}) \approx \sum_{\alpha_d,\dots,\alpha_d} \rho^{(1)}_{\alpha_0,\alpha_1}(x_1) \cdots \rho^{(d)}_{\alpha_{d-1},\alpha_d}(x_d)$. This gives the same function as a multilinear interpolation of the *value* of the discrete approximation (10). However, interpolation of TT cores is the more efficient computation.

The attraction of approximating the PDF in TT format is that the construction of the discrete TT cores $\hat{\rho}^{(k)}$ via alternating TT-cross algorithms [9], [22], basic linear algebra subprograms [11], and evaluation of expectations as in (6) can all be performed with cost that is also at most polynomial in the system dimension d as long the TT ranks remain bounded, which holds for sufficiently regular density functions.

IV. Filtering in TT format

Filtering in the TT format requires implementing the prediction (2) and update (5) steps given in Sec. I-B. As we did with the FVF, the prediction step will be performed by integrating the Fokker–Planck equation (7), using the method in [19]. Operations required in the update step follow the methods in [11].

A. Fokker–Planck equation

We discretize the PDE (7) using the tensor product finite difference (FD) scheme, as follows. For each variable from $(x_1, \dots, x_d) =: \boldsymbol{x}$ we use a uniform grid with n points

$$x_k^{i_k} = -a_k + i_k \frac{2a_k}{n}, \quad i_k = 1, \dots, n, \quad k = 1, \dots, d.$$

on an interval $[-a_k, a_k]$ chosen large enough such that densities are negligible outside.

Treating $\hat{\rho}$ as a vector of length n^d, we write the discretized Fokker–Planck equation as an ODE

$$\frac{d\hat{\rho}}{dt} = H\hat{\rho}(t), \quad H = \sum_{k=1}^{d} \left[-\nabla_{kh} \mathrm{diag}(\hat{f}_k) + \epsilon \Delta_{kh} \right], \tag{11}$$

where $\mathrm{diag}(\hat{f}_k)$ is a diagonal matrix with the elements of $\hat{f}_k$ along the diagonal, ϵ is half the evolution noise variance, i.e., the scaling of the second derivative term in (7) and assuming a diagonal covariance, and

$$\nabla_{kh} = I \otimes \cdots \otimes \nabla_h \otimes \cdots \otimes I, \quad \Delta_{kh} = I \otimes \cdots \otimes \Delta_h \otimes \cdots \otimes I$$

are matrices approximating first and second partial derivatives in variable k. This discretization conserves probability, and for a sufficiently large grid size n it also preserves positivity. Hence, this space-discretized form defines a Markov operator [14].

B. ODE solver

Because the TT storage of the PDF scales linearly in dimension, it is feasible (actually desirable) to include the time variable as one dimension, and represent the PDFs over both state space and the time interval between measurements. This has the advantage that it is natural then to employ an *implicit* time integrator; in this case we use a linear piecewise Chebyshëv (implicit) scheme to integrate the ODE [19].

We choose a basis of Lagrange polynomials $\{L_m(t)\}_{m=1}^M$ centered at Chebyshëv nodes $\{\tau_m\}_{m=1}^M$ and represent the density

$$\hat{\rho}(t) \approx \sum_{m=1}^{M} \hat{\rho}(\tau_m) L_m(t), \quad t \in (0, \tau].$$

We collect the *snapshots* $\hat{\rho}(\tau_m)$ into a vector $\overline{\rho} = [\hat{\rho}(\tau_m)]_{m=1}^M$ of length $n^d M$, which we aim to find. The time derivative operator is discretized in accordance to the spectral approximation theory [23] as a *differentiation matrix* $S = [L'_\ell(\tau_m)]_{m,\ell=1}^M$. Applying this discretization scheme to ODE (11) we obtain the linear system

$$\underbrace{(S \otimes I - I \otimes H)}_{A}\overline{\rho} = \underbrace{(Se_M) \otimes \hat{\rho}(0)}_{f}.$$

Here $\overline{\rho} = \{\hat{\rho}(\tau_m)\}_{m=1}^M$ is the unknown vector of discrete densities, e_M is the vector of size M full of 1's, and I denotes an identity matrix of appropriate size. These algebraic equations are solved in the TT approximation by the Alternating Minimal Energy (AMEn) algorithm [19].

C. Tracking N weakly-coupled pendulums

We now consider estimating the state of N weakly coupled pendulums when we observe the angle of just one pendulum.

N weakly coupled pendulums are described by a $d = 2N$-dimensional kinematic evolution

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \theta_1 \\ \omega_1 \\ \theta_2 \\ \omega_2 \\ \vdots \\ \theta_N \\ \omega_N \end{pmatrix} = \begin{pmatrix} \omega_1 \\ -\sin\theta_1 + \kappa\left[\theta_2 - \theta_1\right] \\ \omega_2 \\ -\sin\theta_2 + \kappa\left[-(\theta_2 - \theta_1) + (\theta_3 - \theta_2)\right] \\ \vdots \\ \omega_N \\ -\sin\theta_N + \kappa\left[-(\theta_N - \theta_{N-1})\right] \end{pmatrix}.$$

The coupling force is similar to the discrete one-dimensional Laplacian acting on the angle variables. We took $\kappa = 0.2$, and simulated data with initial conditions $(\theta_i, \omega_i) = (0.25, 0)$, $i = 1, 2, \ldots, N$. Observations each $t = 0.4$ are of the angle of the first pendulum θ_1 subject to zero-mean additive Gaussian noise with standard deviation 0.2.

1) $N = 2$ weakly-coupled pendulums ($d = 4$): The first example is for 2 pendulums, so $d = 4$, and the angle of the first pendulum is observed, as detailed above. The results for running the FD filter in TT format on this problem are shown in Fig. 9.

The filter is clearly working well, and behaving as expected, with information about the second pendulum being acquired over time. Moreover, the TT ranks of the PDF decrease as the PDF concentrates around the true kinematic variables.

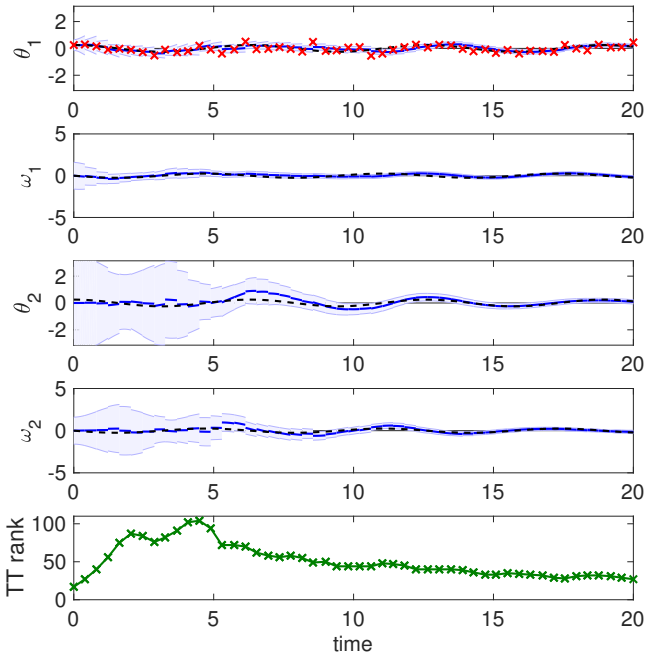

Fig. 9. Filtering for the state of two, lightly-coupled pendulums. Red crosses are observed (noisy) position of the first pendulum. Solid blue line shows the mean of marginal distributions for each state variable, shaded region is two standard-deviation interval, either side of mean. Bottom line indicates the maximal TT ranks of the PDF.

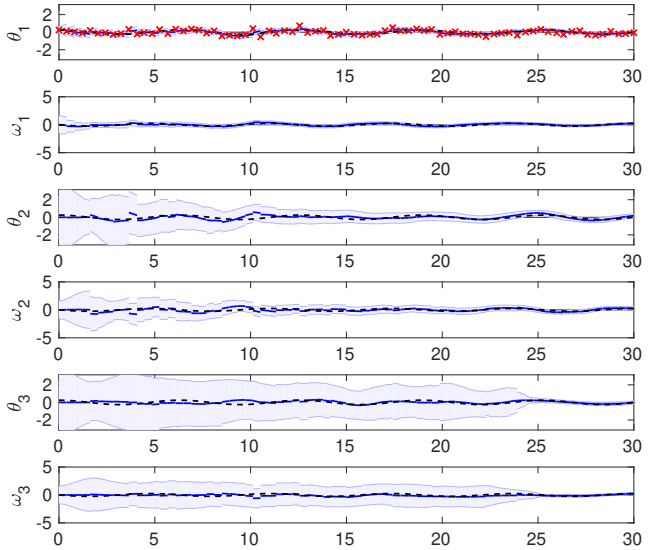

Fig. 10. Filtering for the state of three lightly-coupled pendulums. Red crosses are observed (noisy) position of the first pendulum. Solid blue line shows the mean of marginal distributions for each state variable, shaded region is two standard-deviation interval, either side of mean.

2) $N = 3$ weakly-coupled pendulums ($d = 6$): The analogous results for running the FD filter in TT format on $N = 3$ weakly-coupled pendulums, with the angle of the first pendulum observed, are shown in Fig. 10. It is clear that inference of the third pendulum (Fig. 10) is more difficult than inference for the second, as one would expect, and requires more measurements to recover the third pair of variables with a reasonable confidence.

Figure 11 shows the prior PDF and marginal PDFs for each pendulum at time $t = 6$. It is clear, particularly from the PDF for the third pendulum (right-most panel) that the TT representation can accommodate complex-shaped PDFs.

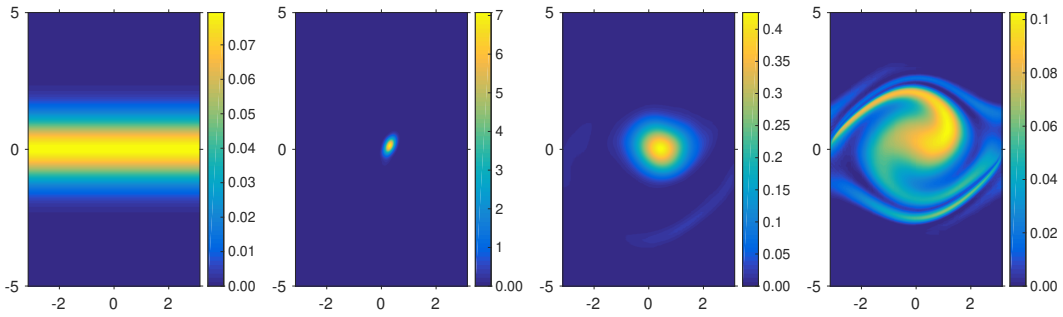

Fig. 11. Initial and transient PDFs generated by filtering for the state of three lightly-coupled pendulums. Shown are the prior distribution and the three marginal distributions at $t = 6$ over pendulums 1, 2, 3 (left to right) in θ_k (horizontal) and ω_k (vertical).

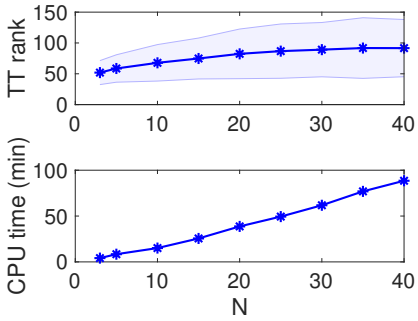

Fig. 12. TT ranks (top) and CPU times of FD filtering in TT format in minutes (bottom) for different numbers of pendulums N. CPU is Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz with 64Gb RAM. Solid blue line shows the mean, shaded region is one standard-deviation interval, either side of mean.

3) TT ranks for N weakly-coupled pendulums ($d = 2N$): We also ran the FD filter in TT format on the same weakly-coupled pendulums problem with various number of pendulums up to $N = 40$, i.e., $d = 80$, to evaluate the scaling of compute cost with dimension. We choose a more informative prior of $\mathrm{MVN}(0, 0.09)$ in each variable. A summary of the TT ranks and compute time is shown in 12. We see that the TT ranks remain bounded for a range of dimensions, and the computing time of filtering grows approximately linearly with the dimension.

V. Conclusions

As is well known [1], [13], and was shown in Sec. I-C, Bayes-optimal filtering for a continuous-time dynamical system with discrete-time observations requires simulating the forward Kolmogorov equation (FKE) that can be written as an initial-value problem in the Fokker–Planck PDE. It is interesting to view existing filters in terms of the density function representation and the PDE solver implemented, perhaps implicitly. For example, Kalman filters use Gaussian representations of PDFs and solve the PDE given by linear, or linearized, system dynamics. The particle filters represent distributions by a set of random point masses, or *particles*, and compute the trajectory of each particle along characteristics. In this paper, we utilized classical PDE solvers from scientific computing, that also exploit *smoothness* of functions for greater efficiency.

We developed two particular filters: the finite-volume filter (FVF) performed well on low-dimensional systems, while the finite-difference filter utilizing a TT representation achieved a higher rate of convergence due to the use of an implicit time integrator and also scaled well to higher dimensions by showing linear scaling of compute cost with dimension, in the examples we computed. Both filters are optimal, i.e., compute essentially exactly with nonlinear dynamics and arbitrary probability distributions, and can represent multi-modal PDFs so are potentially applicable to filtering for multiple target tracking.

We are grateful to an anonymous referee for alerting us to one other paper [13] where tensor decomposition methods (in particular the QTT format) has also been used to implement grid-method filtering, in the continuous-continuous setting, by numerical solution of Duncan-Mortensen-Zakai (DMZ) equation, and the forward Kolmogorov equation arising from the conditioning of DMZ on data. The algorithm in [13] approximates the matrix exponential of the stiffness matrix by a power of the shifted matrix in the QTT format, similarly to the explicit Euler scheme. When a sufficiently high matrix power can be pre-computed accurately (especially in low dimensions), this approach can be faster than the tAMEn method [19] that we use since one does not need to solve linear systems in each step. However, we observed that in higher dimensions the computation of the matrix power in TT can accumulate the error rather quickly, whereas implicit methods (including tAMEn) are more accurate and numerically stable. Further, in filtering applications where preserving positivity of PDFs is desirable, implicit solvers such as tAMEn allow higher-order convergence of the time integrator than explicit integrators that are limited to first-order accuracy, as noted in section II-D.

We conclude that continuous-discrete filtering in the TT format shows great promise for applications where currently only approximate, sub-optimal filters are available.

REFERENCES

[1] A. H. Jazwinski, *Stochastic processes and filtering theory*. New York: Acad. Press, 1970.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Proces.*, vol. 50, no. 2, pp. 174–188, 2002.

[3] R. S. Bucy, "Bayes theorem and digital realizations for non-linear filters," *J. Astronaut. Sci.*, vol. 17, no. 2, pp. 80–94, 1969.

[4] A. Doucet, N. de Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo methods in practice*, ser. Statistics for Engineering and Information Science, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York: Springer-Verlag, 2001, ch. 1, pp. 1–14.

[5] F. Daum, "Mesh-free adjoint methods for nonlinear filters," *Proc. SPIE*, vol. 5913, pp. 59 130D–59 130D–10, 2005.

[6] ——, "Nonlinear filters: beyond the Kalman filter," *IEEE Aero. El. Sys. Mag.*, vol. 20, no. 8, pp. 57–69, Aug. 2005.

[7] K. Kastella and C. Kreucher, "Multiple Model Nonlinear Filtering for Low Signal Ground Target Applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 2, pp. 549–564, APRIL 2005.

[8] P. Dutta and R. Bhattacharya, "Hypersonic State Estimation Using Frobenius-Perron Operator," *J. Guid. Control Dyn.*, vol. 34, no. 2, pp. 325–344, 2011.

[9] I. V. Oseledets and E. E. Tyrtyshnikov, "TT-cross approximation for multidimensional arrays," *Linear Algebra Appl.*, vol. 432, no. 1, pp. 70–88, 2010.

[10] I. Oseledets, "DMRG approach to fast linear algebra in the TT–format," *Comput. Meth. Appl. Math.*, vol. 11, no. 3, pp. 382–393, 2011.

[11] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.

[12] I. Oseledets, "Constructive representation of functions in low-rank tensor formats," *Constr. Approx.*, vol. 37, no. 1, pp. 1–18, 2013.

[13] S. Li, Z. Wang, S. S. T. Yau, and Z. Zhang, "Solving high-dimensional nonlinear filtering problems using a tensor train decomposition method," arXiv preprint 1908.04010, 2019. [Online]. Available: http://arxiv.org/abs/1908.04010

[14] A. Lasota and M. C. Mackey, *Chaos, fractals, and noise*, 2nd ed., ser. Applied Mathematical Sciences. Springer-Verlag, New York, 1994, vol. 97.

[15] H. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics : The Finite Volume Method*. Pearson Education, US, 2007.

[16] R. A. Norton, C. Fox, and M. E. Morrison, "Numerical approximation of the Frobenius–Perron operator using the finite volume method," *SIAM Journal on Numerical Analysis*, vol. 56, no. 1, pp. 570–589, 2018.

[17] C. Fox, R. A. Norton, M. E. Morrison, and T. C. Molteno, "Sequential Bayesian inference for dynamical systems using the finite volume method," in *2017 MATRIX Annals*. Springer, 2019, pp. 13–23.

[18] S. Dolgov, K. Anaya-Izquierdo, C. Fox, and R. Scheichl, "Approximation and sampling of multivariate probability distributions in the tensor train decomposition," *Statistics and Computing*, Nov 2019. [Online]. Available: https://doi.org/10.1007/s11222-019-09910-z

[19] S. V. Dolgov, "A tensor decomposition algorithm for large ODEs with conservation laws," *Computational Methods in Applied Mathematics*, vol. 19, pp. 23–38, 2019. [Online]. Available: http://arxiv.org/abs/1403.8085

[20] G. H. Golub and C. F. Van Loan, *Matrix Computations, forth edition*. Johns Hopkins University Press, 2013.

[21] S. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin, "A theory of pseudoskeleton approximations," *Linear Algebra Appl.*, vol. 261, no. 1–3, pp. 1–21, 1997.

[22] S. Dolgov and D. Savostyanov, "Parallel cross interpolation for high-precision calculation of high-dimensional integrals," *Comp. Phys. Comm.*, vol. 246, p. 106869, 2019.

[23] L. N. Trefethen, *Spectral methods in MATLAB*. Philadelphia: SIAM, 2000.