Efficient Spectral Differentiation in Grid-Based Continuous State Estimation

J. Matoušek, J. Duník, M. Brandner

***Abstract*—This paper deals with the state estimation of stochastic models with continuous dynamics. The aim is to incorporate spectral differentiation methods into the solution to the Fokker-Planck equation in grid-based state estimation routine, while taking into account the specifics of the field, such as probability density function (PDF) features, moving grid, zero boundary conditions, etc. The spectral methods, in general, achieve very fast convergence rate of $\mathcal{O}(c^N)(O < c < 1)$ for analytical functions such as the probability density function, where N is the number of grid points. This is significantly better than the standard finite difference method (or midpoint rule used in discrete estimation) typically used in grid-based filter design with convergence rate $\mathcal{O}(\frac{1}{N^2})$. As consequence, the proposed spectral method based filter provides better state estimation accuracy with lower number of grid points, and thus, with lower computational complexity.**

***Index Terms*—State estimation, transition probability matrix, Chapman-Kolmogorov equation, Fokker-Planck equation, point-mass filter, spectral differentiation, spectral methods.**

I. Introduction

The goal of this paper is to introduce the usage of spectral methods to state estimation, and to show how the spectral methods can benefit from the efficient grid-based point mass filter formulation proposed in [1]–[3].

The continuous state estimation starts with a model described by the stochastic differential equation (SDE), and stochastic algebraic equation

$$d\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t)dt + \mathbf{Q}d\mathbf{w}(t) \tag{1}$$

$$\mathbf{z}_{t_k} = \mathbf{h}(\mathbf{x}_{t_k}) + \mathbf{v}_{t_k}, \tag{2}$$

where t denotes the time, the vector $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ represents the *unknown* state of the system, $\mathbf{z}_{t_k} \in \mathbb{R}^{n_z}$ the *known* measurement, and $t_k = t_0, t_1, \ldots, T$ is a discrete time instant, when the measurement arrived. The matrix $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ and the function $\mathbf{h}(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_z}$ are known, $\mathbf{Q} \in \mathbb{R}^{n_x \times n_x}$ is a known diffusion coefficient, $\mathbf{w}(t)$ is the state noise modelled by the Brownian motion with the normally distributed increment with the covariance matrix $E[d\mathbf{w}(t)(d\mathbf{w}(t))^T] = \mathbf{I}_{n_x}dt$, and $\mathbf{v}_{t_k}$ is the measurement noise with the known PDF $p(\mathbf{v}_{t_k})$. The state and measurement noise random variables are supposed to be independent mutually and of the *known* initial state $\mathbf{x}_0$ with the known PDF $p(\mathbf{x}_0)$.

Authors are with Department of Cybernetics, Pilsen, Czech Republic. E-mails: e-mail: {dunikj,matoujak}@kky.zcu.cz (J. Duník, J. Matoušek), and Department of Mathematics, Faculty of Applied Sciences, University of West Bohemia, brandner@kma.zcu.cz (M. Brandner).

The aim of the state estimation (or filtering) is to compute the PDF of the state $\mathbf{x}(t)$ conditioned on all past measurements $\mathbf{z}^{t_k} = [\mathbf{z}_{t_0}, \mathbf{z}_{t_1}, \ldots, \mathbf{z}_{t_k}]$. The sought conditional PDF can be either filtering $p(\mathbf{x}(t_k)|\mathbf{z}^{t_k})$, which is computed in the filter *measurement update* step, or predictive $p(\mathbf{x}(t)|\mathbf{z}^{t_k}), t > t_k$, which is computed in the filter *time update* step.

A. Measurement Update

The solution to the measurement update providing the filtering PDF is given by the Bayes' rule

$$p(\mathbf{x}_{t_k}|\mathbf{z}^{t_k}) = \frac{p(\mathbf{x}_{t_k},\mathbf{z}_{t_k}|\mathbf{z}^{t_{k-1}})}{p(\mathbf{z}_{t_k}|\mathbf{z}^{t_{k-1}})} = \frac{p(\mathbf{x}_{t_k}|\mathbf{z}^{t_{k-1}})p(\mathbf{z}_{t_k}|\mathbf{x}_{t_k})}{p(\mathbf{z}_{t_k}|\mathbf{z}^{t_{k-1}})}, \tag{3}$$

where $p(\mathbf{x}_{t_k}|\mathbf{z}^{t_{k-1}})$ is the predictive[1] PDF, $p(\mathbf{z}_{t_k}|\mathbf{x}_{t_k})$ is the measurement PDF obtained from (2), and the normalization coefficient is given by $p(\mathbf{z}_{t_k}|\mathbf{z}^{t_{k-1}}) = \int p(\mathbf{x}_{t_k}|\mathbf{z}^{t_{k-1}})p(\mathbf{z}_{t_k}|\mathbf{x}_{t_k})d\mathbf{x}_{t_k}$.

B. Time Update

The solution to the time update, w.r.t. the SDE (1), is given by the Fokker-Planck equation (FPE) describing the time evolution of the filtering PDF to predictive PDF

$$\frac{\partial p_t(\mathbf{x}|\mathbf{z}^{t_k})}{\partial t} = -\nabla \cdot \left(\mathbf{A}\mathbf{x}p_t(\mathbf{x}|\mathbf{z}^{t_k})\right) + \frac{1}{2}\nabla \cdot \left(\mathbf{Q}\nabla p_t(\mathbf{x}|\mathbf{z}^{t_k})\right), \tag{4}$$

where $t \in (t_k, t_{k+1})$, the operator ∇ denotes the gradient w.r.t. $\mathbf{x}$, and $\nabla\cdot$ stands for the divergence. Further, when possible, the conditional PDF is written simply as $p_t(\mathbf{x})$.

In (4), the first right-hand side term is named *hyperbolic* and it describes the *advection* of the PDF tied to the state dynamics. The second term is named *parabolic* and it describes the *diffusion* caused by the state noise. The standard FPE (4) holds for the Gaussian state noise only [4].

II. FPE Efficient Solution Prerequisites

The FPE is exactly solvable for a narrow set of the SDEs (1). Within the set, it is possible to find the linear SDEs, for which the solution to the FPE (4) leads to the prediction step of the Kalman-Bucy filter [5], or SDEs of special form that are solved by e.g. Bene, Daum, and Wong and Yau filters [6]. For remaining SDEs, the FPE is not

[1]The predictive PDF $p(\mathbf{x}_{t_0}|\mathbf{z}^{t-1})$ is equal to the initial one $p(\mathbf{x}_0)$.

exactly solvable and need to be solved either using model linearisation or numerically on a grid of points [5], [7], [8].

Before the numerical solution is discussed, the FPE is rewritten into suitable form allowing simple application of fast frequency-based numerical methods. The changes are twofold, first, a diagonalisation of matrix $\mathbf{Q}$ is performed, second, an advection is solved using Lagrangian approach.

A. Diagonalization

First step that greatly reduces the complexity of the advanced efficient numerical algorithms is a diagonalisation of the $\mathbf{Q}$ in (1) and subsequently in (4). Instead of solving the FPE for the model with non-diagonal $\mathbf{Q}$, an alternative diagonalized FPE can be solved

$$\frac{\partial p_t(\bar{\mathbf{x}})}{\partial t} = -\nabla_{\bar{\mathbf{x}}} \cdot \left(\bar{\mathbf{A}}\bar{\mathbf{x}}\ p_t(\bar{\mathbf{x}})\right) + \frac{1}{2}\nabla_{\bar{\mathbf{x}}} \cdot \left(\nabla^T_{\bar{\mathbf{x}}} p_t(\bar{\mathbf{x}})\right), \tag{5}$$

where the following substitution was done, $\mathbf{Q} = \mathbf{G}\mathbf{G}^T$, $\bar{\mathbf{x}} = \mathbf{G}^{-1}\mathbf{x}$, $\bar{\mathbf{A}} = \mathbf{G}^{-1}\mathbf{A}\mathbf{G}$, and $\nabla_{\bar{\mathbf{x}}} = [\frac{\partial}{\partial \bar{\mathbf{x}}(1)} \cdots \frac{\partial}{\partial \bar{\mathbf{x}}(n_x)}]$. Then the estimation algorithm is run in the state space $\bar{\mathbf{x}}$. If the first two moments (i.e. the estimate and its uncertainty) for the original FPE solution are needed they can be calculated as $\mathsf{E}[\mathbf{x}] = \mathbf{G}\mathsf{E}[\bar{\mathbf{x}}]$, $\mathsf{cov}[\mathbf{x}] = \mathbf{G}\ \mathsf{cov}[\bar{\mathbf{x}}]\ \mathbf{G}^{-1}$. For simplicity it is further assumed that $\mathbf{Q}$ is diagonal from the very beginning, which is true for many models such as the model in the numerical illustration section of this paper.

B. Advection solution

The second step applies a Lagrangian approach to solve the advection part of the FPE $\nabla p_t(\mathbf{x})\,(\mathbf{A}\mathbf{x})$ (4). This part is problematic because the product of $p_t(\mathbf{x})$ and $\mathbf{A}\mathbf{x}$ depends on the actual state, which prevents efficient solution in frequency domain. This part can be disposed of by using - in a sense - a Lagrangian approach [9, pp. 7].

Let every point of the state-space $\mathbf{x} \in \mathbb{R}^{n_x}$ be moving according to[2] $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$. Then, the FPE advection part (4) taking into account the movement reads [10]

$$\nabla \cdot \left(\mathbf{A}\mathbf{x}\ p_t(\mathbf{x})\right) = \nabla p_t(\mathbf{x})\ \dot{\mathbf{x}} + \mathrm{trace}(\mathbf{A}) p_t(\mathbf{x}), \tag{6}$$

where $\mathrm{trace}(\mathbf{A})$ is the trace of the matrix $\mathbf{A}$.

The FPE to be solved is then

$$\frac{\partial p_t(\mathbf{x})}{\partial \mathbf{v}} = -\,\mathrm{trace}(\mathbf{A}) p_t(\mathbf{x}) + \frac{1}{2}\nabla \cdot \left(\mathbf{Q}\left(\nabla^T p_t(\mathbf{x})\right)\right), \tag{7}$$

where $\mathbf{v} = \begin{bmatrix}\dot{\mathbf{x}} & 1\end{bmatrix}$, and

$$\frac{\partial p_t(\mathbf{x})}{\partial \mathbf{v}} = \mathbf{v}\left[\nabla p_t(\mathbf{x})\ ,\quad \tfrac{\partial p_t(\mathbf{x})}{\partial t}\right]^T. \tag{8}$$

Compared to the FPE (4), where the PDF is differentiated w.r.t. the time t, in (7) it is differentiated w.r.t. observer, which is moving in time. As will be seen, the remaining advection term $\mathrm{trace}(\mathbf{A})p_t(\mathbf{x})$ can be easily handled.

[2]Movement of the PDF caused by the advection is compensated by the movement of the observer. Imagine you are one particle of water in the river, in a simplified way, from your point of perspective, the water around you is not moving.

III. Grid-Based Estimation

In this section a grid-based estimation for continuous dynamics is briefly introduced, starting with usually used approximation to the PDFs. After that two grid-based estimation methods are presented; *(i)* standard basic finite difference method in state domain, and *(ii)* is its efficient formulation in frequency domain.

Methods, are presented in their one-dimensional form for better understanding of the underlying concepts. Multidimensional variants can be found in [1]–[3].

The reason for using grid-based methods over meshless methods such as particle methods is that grid based methods offer deterministic results and are believed to be more robust [11].

A. Density Approximation

The grid-based filters are based on an approximation of a conditional PDF $p_t(\mathbf{x})$ by a *piece-wise constant* point-mass density (PMD) $\hat{p}_t(\mathbf{x};\boldsymbol{\xi}_t)$ defined at the set of the discrete grid points[3] $\boldsymbol{\xi}_t = [\boldsymbol{\xi}_t^{(:,0)}, ..., \boldsymbol{\xi}_t^{(:,N-1)}]$, $\boldsymbol{\xi}_t^{(:,i)} \in \mathbb{R}^{n_x}$, as follows

$$\hat{p}_t(\mathbf{x};\boldsymbol{\xi}_k) \triangleq \sum_{i=0}^{N-1} P_t(\boldsymbol{\xi}_t^{(:,i)}) S_t\{\mathbf{x};\boldsymbol{\xi}_t^{(:,i)},\boldsymbol{\delta}_t\}, \tag{9}$$

with

- $P_t(\boldsymbol{\xi}_t^{(:,i)}) = c_t \tilde{P}_t(\boldsymbol{\xi}_t^{(:,i)})$, where $\tilde{P}_t(\boldsymbol{\xi}_t^{(:,i)}) = p_t(\boldsymbol{\xi}_t^{(:,i)})$ is the value of the conditional PDF $p_t(\mathbf{x})$ evaluated at the i-th grid point $\boldsymbol{\xi}_t^{(:,i)}$ (also called PMD point weight), $c_t = \delta_t \sum_{i=1}^{N} \tilde{P}_t(\boldsymbol{\xi}_t^{(:,i)})$ is a normalisation constant, and δ_t is the volume of the i-th point neighbourhood,
- $\boldsymbol{\delta}_t$ defines a (hyper-) rectangular neighbourhood[4] of a grid point, where the PDF $p_t(\mathbf{x})$ is assumed to be constant and has value $P_t(\boldsymbol{\xi}_t^{(:,i)})$,
- $S_t\{\mathbf{x};\boldsymbol{\xi}_t^{(:,i)},\boldsymbol{\delta}_t\}$ is the *selection* function defined as

$$S_t\{\mathbf{x};\boldsymbol{\xi}_t^{(:,i)},\boldsymbol{\delta}_t\} = \begin{cases} 1, \text{if } |\mathbf{x}^{(j)} - \boldsymbol{\xi}_t^{(j,i)}| \le \frac{\boldsymbol{\delta}_t^{(j)}}{2} \forall j, \\ 0, \text{otherwise.} \end{cases} \tag{10}$$

- $N = (N_{\text{pa}})^{n_{\mathbf{x}}}$ is the total number of grid points and N_{pa} means the number of discretisation points per axis (pa) for grid with boundaries aligned with the state-space axes.

B. Finite Difference Method

The finite difference method (FDM) can be considered as the baseline approach typically used in the continuous grid-based state estimation, where the continuous conditional PDF support is approximated by the grid points in which the differences are computed to solve the FPE.

[3]MATLAB® / Python® style notation is used throughout the paper for simple comparison with published codes, where $(j+1)$-th element of vector $\mathbf{x}_t$ is denoted as $\mathbf{x}_t^{(j)}$ and element of matrix $\boldsymbol{\xi}_t$ at $(i+1)$-th row and $(j+1)$-th column is denoted as $\boldsymbol{\xi}_t^{(j,i)}$. The indexing is from 0 as it allows transparent notation for spectral differentiation theory.

[4]Assumed to be same for each grid point, i.e. the grid is equidistant.

The number of possible FDM numerical schemes is vast, and their overview can be e.g. found in [12]. The baseline approach in the state estimation is an explicit scheme with central difference for diffusion (upwind for advection). The FPE (7) is solved from t_k to t_{k+1} using l numerical steps with length Δt, i.e., $l = \frac{t_{k+1}-t_k}{\Delta t}$.

For the FPE (7), and the enforced grid movement

$$\dot{\xi}_n^{(j)} = A\xi_n^{(j)}, \forall j, \tag{11}$$

an explicit 1D scheme can be derived, using shorthand notation $P_n^{(j)} = P_{t_k+n\Delta t}(\xi_{t_k+n\Delta t}^{(j)})$, as

$$P_{n+1}^{(j)} = P_n^{(j)} - AP_n^{(j)} + \frac{\Delta t}{2\delta_n^2} Q\left(P_n^{j+1} - 2P_n^{(j)} + P_n^{j-1}\right), n = 0, 1, \ldots, l, \tag{12}$$

where $\delta_n = e^{A\Delta t}\delta_{n-1}$. For multi-dimensional case, a derivative in each direction is approximated by the difference, see e.g. [12].

C. Efficient Formulation

In standard formulation, to calculate the advection part of the FPE (4) a product of dynamics Ax and PDF $p_t(x)$ has to be done. In practice this equals to Hadamard product of two vectors. Unfortunately, Hadamard product in the time domain becomes convolution in the frequency domain. Therefore, for each numerical time step, the PDF derivative has to be converted back to the time domain, the product calculated and the result converted back to the frequency domain, leading to computational overhead. However, the efficient FPE formulation (7) circumvents this issue as it solves the advection term by a grid movement.

The finite difference solution (12) can be rewritten to a matrix form, forming a tridiagonal matrix

$$\mathbf{F}_{\text{diff}}(t) = \begin{bmatrix} b_t & a_t & & & \\ a_t & b_t & a_t & & \\ & a_t & b_t & a_t & \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \tag{13}$$

where $a_t = \frac{Q}{2\delta_t^2}\frac{\Delta t}{}$, and $b_t = 1 - \frac{Q\Delta t}{\delta_t^2} - \Delta t A$. Note that the coefficients a_t, b_t are row-independent due to the FPE with enforced grid movement (7).

The resulting prediction, i.e., a numerical solution to the FPE (7), from the time instant k to $k+1$, becomes

$$P_{t_{k+1}}^{(:)} = \underbrace{\mathbf{F}_{\text{diff}}(t_k + l\Delta t)\cdots\mathbf{F}_{\text{diff}}(t_k + \Delta t)\mathbf{F}_{\text{diff}}(t_k)}_{\mathbf{T}_k} P_{t_k}^{(:)}, \tag{14}$$

where the shorthand notation $P_{t_k}^{(:)} = \left[P_{t_k}^{(0)}, P_{t_k}^{(1)}, \ldots, P_{t_k}^{(N-1)}\right]^T$ is used.

It can be shown that thanks to the grid movement (11), the eigenvector matrix $\mathbf{R}$ of time dependent $\mathbf{F}_{\text{diff}}(t)$ are constant, while its eigenvalues λ_t are time-dependent. Thus $\mathbf{T}_{t_k}$ can be, efficiently, calculated using the eigenvalue and eigenvector form of $\mathbf{F}_{\text{diff}}(t)$ as

$$\mathbf{T}_{t_k} = \mathbf{R} \underbrace{\left(\mathbf{\Lambda}(t_k + (l-1)\Delta t) \odot \cdots \odot \mathbf{\Lambda}(t_k + \Delta t) \odot \mathbf{\Lambda}(t_k)\right)}_{\mathbf{\Lambda}_{t_k}} \mathbf{R}^{-1}, \tag{15}$$

where $\mathbf{\Lambda}_k$ is a matrix with eigenvalues $\lambda_t^{(j)}$ on diagonal. Because the matrix $\mathbf{F}_{\text{diff}}$ (13) is Toeplitz, its eigenvalues are [13]

$$\lambda_t^{(j-1)} = b_t + 2a_t \cos\left(\frac{j\pi}{N+1}\right), j = 1, \ldots, N, \tag{16}$$

and eigenvectors

$$\mathbf{r}^{(j-1)} = \begin{bmatrix} \sin(\frac{1j\pi}{N+1}) \\ \vdots \\ \sin(\frac{Nj\pi}{N+1}) \end{bmatrix}, \ j = 1, ..., N. \tag{17}$$

For the eigenvector matrix, it also holds that

$$\mathbf{R}^{-1} = \frac{2}{N+1}\mathbf{R}. \tag{18}$$

Therefore, calculation of an arbitrary element $\mathbf{T}_{t_k}^{(j-1,i-1)}$ reads

$$\mathbf{T}_{t_k}^{(j-1,i-1)} = \frac{2}{N+1}\sum_{\ell=1}^{N} \mathbf{\Lambda}_{t_k}^{(\ell,\ell)} \sin\left(\frac{i\ell\pi}{N+1}\right)\sin\left(\frac{j\ell\pi}{N+1}\right), \tag{19}$$

where $i = 1, \ldots, N$ and $j = 1, \ldots, N$. Note that calculation of the element $\mathbf{T}_{t_k}^{(j-1,i-1)}$ is independent of any other element compared to the standard FDM-based (12) solution. Thanks to this special form, the *fast sine transform* $\mathcal{S}$ can be used [14] to calculate prediction as

$$P_{t_{k+1}}^{(:)} = \mathcal{S}\left(\text{diag}(\mathbf{\Lambda}_{t_k}) \odot \mathcal{S}(P_{t_k}^{(:)})\right), \tag{20}$$

where $\odot$ is Hadamard product. For further information and multidimensional derivation please refer to [1]–[3].

D. Computational Complexity and Convergence Rate

The computational complexity of predictive PDF calculation in grid-based method was reduced from $\mathcal{O}(N^2)$ in (12) to $\mathcal{O}(N \log N)$ in (20). However, the convergence rate is still $\mathcal{O}(\frac{1}{N^2})$.

Thus, the aim of this paper is to keep the reduced computational complexity and also enhance the convergence rate in space, i.e., to improve FPE solution with reduced number of grid points N.

IV. Spectral differentiation

Spectral differentiation (SD) is a method of calculating derivative of a function in a frequency domain, such as derivatives present in the FPE (7). The main advantage of the spectral methods is very fast convergence rate $\mathcal{O}(c^N)(O < c < 1)$ for analytic functions [15, pp. 41].

Basic idea of the SD in this case is, instead of calculating the conditional PMD differences in the state-space domain, to interpolate the Fourier transformation of the PMD and calculate its derivative analytically.

Let a one dimensional PDF $p_t(x)$ be sampled on a grid with N grid points $\xi_t^{(j)}$, where N is assumed to be even, and let $L_t = N\delta_t$ be a grid size. Then, the PMD can be expressed in a frequency domain using discrete Fourier transform

$$\mathcal{P}_t^{(s)} = \frac{1}{N}\sum_{j=0}^{N-1} P_t^{(j)} \exp^{-\frac{2\pi i}{N}js}, \tag{21}$$

where $s = 0, \ldots, N-1$ and i is imaginary unit.

To compute derivatives of (21) analytically, a continuous interpolation in the frequency domain has to be defined. A unique "minimal-oscillation" trigonometric interpolation of order N is [16]

$$\begin{aligned} p_t(x) =& \mathcal{P}_t^{(0)} + \sum_{0<s<N/2}\left(\mathcal{P}_t^{(s)} e^{\frac{2\pi i}{L_t}sx} + \mathcal{P}_t^{(N-s)} e^{-\frac{2\pi i}{L_t}sx}\right) + \\ & \mathcal{P}_t^{(N/2)} \cos\left(\frac{\pi}{L_t}Nx\right). \end{aligned} \tag{22}$$

Interpolated PMD in frequency domain (22) can easily be differentiated w.r.t. state, $\frac{\partial p_t^{(j)}}{\partial x} = p_t'^{(j)}$ and evaluated at each grid point

$$\begin{aligned} p_t'^{(j)} &= \sum_{0<s<N/2} \underbrace{\frac{2\pi i}{L_t}s}_{c_t'} \left(\mathcal{P}_t^{(s)} e^{\frac{2\pi i}{N}js} + \mathcal{P}_t^{(N-s)} e^{-\frac{2\pi i}{N}js}\right) \\ &= \sum_{s=0}^{N-1} \mathcal{P}_t'^{(s)} \exp^{\frac{2\pi i}{N}js}. \end{aligned} \tag{23}$$

The second derivative $\frac{\partial^2 p_t^{(j)}}{\partial x^2} = p_t''^{(j)}$, needed for the diffusion solution, is then given as

$$\begin{aligned} p_t''^{(j)} =& -\sum_{0<s<N/2} \underbrace{\left(\frac{2\pi i}{L_t}s\right)^2}_{c_t''} \left(\mathcal{P}_t^{(s)} e^{\frac{2\pi i}{N}js} + \mathcal{P}_t^{(N-s)} e^{-\frac{2\pi i}{N}js}\right) \\ & - \left(\frac{\pi}{L_t}N\right)^2 \mathcal{P}_t^{(N/2)}(-1)^j \\ =& \sum_{s=0}^{N-1} \mathcal{P}_t''^{(s)} \exp^{\frac{2\pi i}{N}js}, \end{aligned} \tag{24}$$

Applying the derivatives in multiple dimensions is done by applying successive one dimensional fast Fourier transforms and differentiations. This will be shown later in algorithmic form.

A. Illustration of Differentiation

In Figure 1, an error of the time update step using the FPE (7) for model (1) without advection term (for the sake of simplicity) can be seen for the filtering PDF $p(\mathbf{x}_{t_k}|\mathbf{z}^{t_k})$ in the form

- the Gaussian mixture with three components,
- the Gaussian PDF.

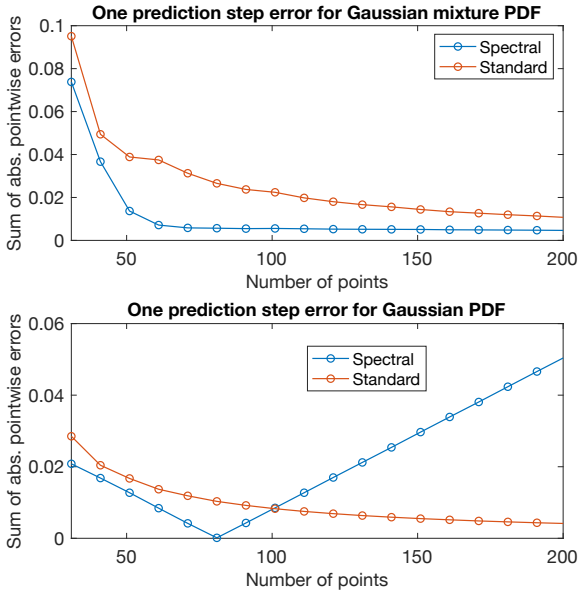

Figure 1: Time update accuracy: convergence for Gaussian PDF and Gaussian mixture PDF.

The fast convergence of the spectral method compared to standard FDM is evident. However, in the case that underlying PDF's are Gaussian a strange behaviour is observed, where the accuracy starts deteriorate with more points. Similar behaviour can be seen for example in [17] without any commentary. Note that, this is not caused by violating the CourantFriedrichsLewy criterion because similar behaviour can be seen in the error of derivative approximation itself.

Our hypothesis is that this behaviour is caused by the trigonometric interpolation as it assumes the function is periodic, please see next section for details.Therefore, in future research we would like to try using the periodic $sinc$ function for interpolation [15, pp. 20].

The trigonometric interpolation was used nonetheless as it is simple and shows good performance, as the considered model with advection (1), (2) is nonlinear and the underlying conditional PDFs are, therefore, inherently non-Gaussian.

Illustration of the errors of the calculated predictive PDF $p_{t_{k+1}}(\mathbf{x}|\mathbf{z}^{t_k})$ for $N = 80$ and $N = 200$ is shown in Figures 2 and 3, respectively, for both filtering densities. It can be seen that for $N = 80$, the SD-based FPE solution provides significantly better prediction for both filtering PDFs. However, for $N = 200$ the accuracy of the prediction using the SD-based FPE solution is worse for the Gaussian filtering PDF as discussed above.

B. Notes

Considered SD method is intended for differentiation of the periodic functions, however, PDFs are not mathematically periodic. Fortunately, when the grid is well designed, the underlying PDF should always be near zero at the boundaries of the grid in such case it can be viewed as periodic in practice [15, pp. 24].

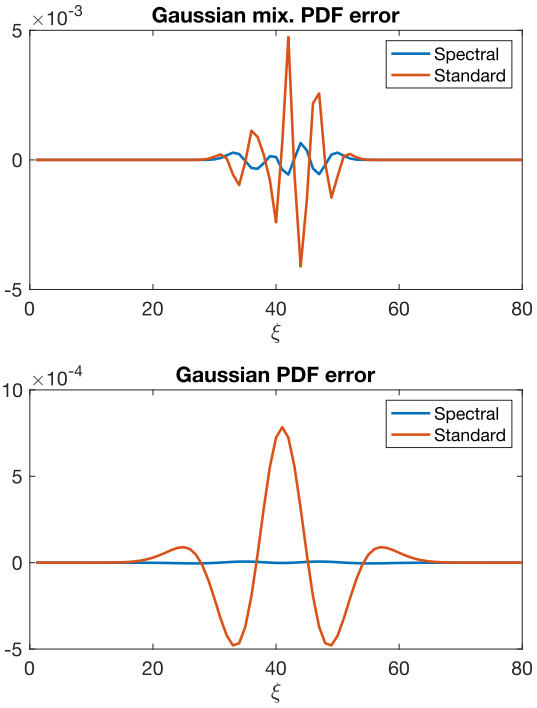

Figure 2: Predictive PMD error for $N = 80$.

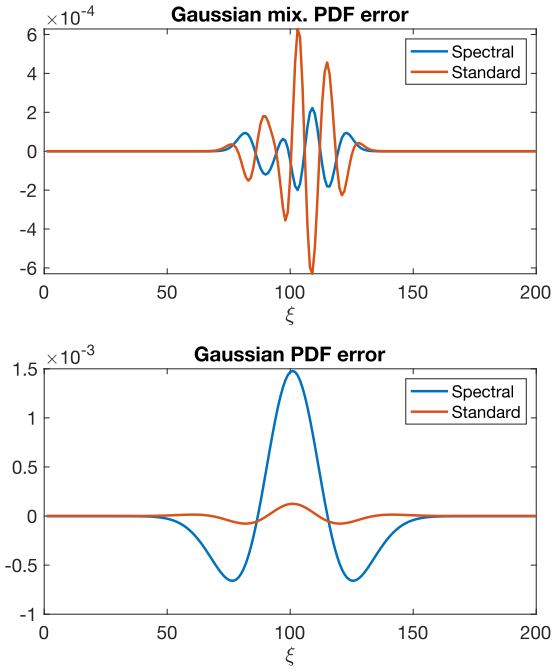

Figure 3: Predictive PMD error for $N = 200$.

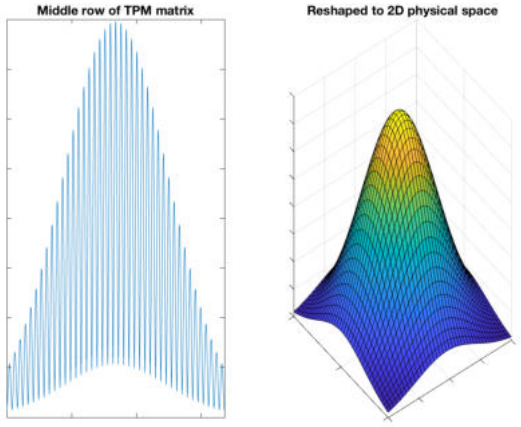

Figure 4: Computational to physical space re-shaping for 2D space and Gaussian noise.

As an alternative for non-periodic functions that are not near zero at the boundaries of the considered domain, a Chebyshev interpolation and differentiation can be used [15, pp. 41]. However, the grid points have to be Chebyshev nodes, such a grid has complex structure and is not suitable for state estimation.

There is a number of other interpolations and approaches that can be use in spectral differentiation, but for the sake of this paper, which has aim to prove the superiority of spectral methods for state estimation in general, the described approach was chosen.

V. Spectral Based Estimation

After the advection was solved by Lagrangian approach and grid movement (7), what remains is to treat the diffusion term

$$\frac{1}{2}\nabla \cdot \left(\mathbf{Q}\nabla p_t(\mathbf{x}(t)\right), \tag{25}$$

where second derivative of $p_t(\mathbf{x})$ is to be calculated. Because $\mathbf{Q}$ is constant and diagonal, the product with $p_t(\mathbf{x})$ can be easily done in a frequency domain.

For simplicity the equations are derived for $n_x = 1$ and the extension for arbitrary n_x is given in the form of the algorithm.

A. Numerical Time Stepping with Spectral Methods

The coefficients to calculate the second derivate, from (24), are then

$$c_t'' = -\left(\frac{2\pi}{L_t}\text{fftshift}\left(-\frac{N_{\text{pa}}}{2} : 1 : (\frac{N_{\text{pa}}}{2} - 1)\right)\right)^2, \tag{26}$$

where MATLAB function `fftshift` shifts the zero frequency component to the middle, and colon follows MATLAB notation. The numerical time stepping including the trace can be, for simplicity (as our main aim here is to

show the spectral method), done using Euler discretisation time step as [18]

$$\frac{1}{\Delta_t}\left(\mathcal{P}_{n+1}-\mathcal{P}_n\right) = -\frac{q}{2}c_n''\odot\mathcal{P}_{n+1}+\operatorname{trace}(\mathbf{A})\mathcal{P}_n$$
$$\Leftrightarrow \mathcal{P}_{n+1} = \left(\mathcal{P}_n(1+\operatorname{trace}(\mathbf{A}))\right)\oslash\left(1-\frac{q}{2}\Delta_t c_n''\right)$$
$$= \mathcal{P}_n(1+\Delta_t\operatorname{trace}(\mathbf{A}))\oslash\psi_n,$$

where $\oslash$ is a Hadamard division.

B. Spectral Methods State Estimation Algorithm

The point-mass filter prediction step based on the introduced SD method and Euler stepping is summarised in the Algorithm below for an arbitrary state dimension n_x.

Algorithm: Efficient spectral point-mass filter weight update

1) *Calculate fast Fourier transform of the filtering PMD weights $P_{t_k|t_k}$ dimension by dimension*: $\widetilde{\mathcal{P}}_{t_k|t_k} = ...\mathcal{F}_{x_2}(\mathcal{F}_{x_1}(P_{t_k|t_k}))$
2) *Calculate coefficients for gradient*:
$$c_{t_k}^{\prime\prime(1,:)} = -\left(\frac{2\pi}{L_{t_k}^{(1)}}\mathit{fftshift}\left(-\tfrac{N_{\text{pa}}}{2}:1:(\tfrac{N_{\text{pa}}}{2}-1)\right)\right)^2$$
$$c_{t_k}^{\prime\prime(2,:)} = -\left(\frac{2\pi}{L_{t_k}^{(2)}}\mathit{fftshift}\left(-\tfrac{N_{\text{pa}}}{2}:1:(\tfrac{N_{\text{pa}}}{2}-1)\right)\right)^2$$
$$\vdots$$
3) *Calculate (generally) tensor Ψ_{t_k} as a product $\psi_{t_k}\odot\psi_{t_k+\Delta_t}\odot\psi_{t_k+2\Delta_t}\cdots\odot\psi_{t_{k+1}}$ where*:
$$\psi_{t_k}^{(i,j,\ldots)} = \frac{1}{1-0.5c_{t_k}^{\prime\prime(1,i)}Q^{(1,1)}\Delta_t-0.5c_{t_k}^{\prime\prime(2,j)}Q^{(2,2)}\Delta_t-\cdots}$$
4) *Calculate the updated weights*: $\mathcal{P}_{t_{k+1}|t_k} = \Psi_{t_k}\odot\mathcal{P}_{t_k|t_k}(1+\Delta_t\operatorname{trace}(\mathbf{A}))^l$
5) *Calculate inverse fast Fourier transform of $\mathcal{P}_{t_{k+1}|t_k}$*
6) *Normalize*

Note that, in the algorithm, the terms with $\sim$ overhead are reshaped to physical space as shown in Figure 4 as discussed in [1]–[3], and $L_{t_k}^{(m)}$ is the size of the grid for m-th dimension.

VI. Numerical Illustration

A model of dynamic coordinated turn with known turn rate α was used [19]. A four-dimensional state $\mathbf{x}_k = [p_x\ v_x\ p_y\ v_y]$, which describes the horizontal position (p_x, p_y) $[m]$ and velocity (v_x, v_y) $[m/s]$ of the vehicle.

A continuous dynamics model reads

$$\mathbf{A} = \begin{bmatrix}0&1&0&0\\0&0&0&-\alpha\\0&0&0&1\\0&\alpha&0&0\end{bmatrix}, \tag{27}$$

$$\mathbf{Q} = \begin{bmatrix}0&0\\1&0\\0&0\\0&1\end{bmatrix}, \tag{28}$$

where $\alpha = 30^\circ$ is the known turn rate.

An analytically derived discrete dynamics model, used for comparison is,

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k \tag{29}$$

$$\mathbf{F} = \begin{bmatrix}1&\frac{\sin(\alpha T_s)}{\alpha}&0&\frac{\cos(\alpha T_s)-1}{\alpha}\\0&\cos(\alpha T_s)&0&-\sin(\alpha T_s)\\0&\frac{1-\cos(\alpha T_s)}{\alpha}&1&\frac{\sin(\alpha T_s)}{\alpha}\\0&\sin(\alpha T_s)&0&\cos(\alpha T_s)\end{bmatrix} \tag{30}$$

$$p(\mathbf{x}_0)\sim\mathcal{N}\left\{\mathbf{x}_0;\begin{bmatrix}36569\\50\\55581\\50\end{bmatrix},\begin{bmatrix}90&0&0&0\\0&160&0&0\\0&0&5&0\\0&0&0&5\end{bmatrix}\right\}, \tag{31}$$

$$p(\mathbf{w}_k)\sim\mathcal{N}\{\mathbf{w}_k;\mathbf{0},\mathbf{Q}_d\}, \tag{32}$$

(33)

$$\mathbf{Q}_d = \begin{bmatrix}\frac{2(\alpha T_s-\sin(\alpha T_s))}{\alpha^3}&\frac{1-\cos(\alpha T_s)}{\alpha^2}&0&\frac{\alpha T_s-\sin(\alpha T_s)}{\alpha^2}\\\frac{1-\cos(\alpha T_s)}{\alpha^2}&T&\frac{-\alpha T_s-\sin(\alpha T_s)}{\alpha^2}&0\\0&\frac{-\alpha T_s-\sin(\alpha T_s)}{\alpha^2}&\frac{\alpha T_s-\sin(\alpha T_s)}{\alpha^2}&0\\\frac{\alpha T_s-\sin(\alpha T_s)}{\alpha^2}&0&0&\frac{2(\alpha T_s-\sin(\alpha T_s))}{\alpha^3}\\0&-\frac{\alpha T_s-\sin(\alpha T_s)}{\alpha^2}&\frac{1-\cos(\alpha T_s)}{\alpha^2}&0\\\frac{\alpha T_s-\sin(\alpha T_s)}{\alpha^2}&0&0&\frac{1-\cos(\alpha T_s)}{\alpha^2}\\0&\frac{1-\cos(\alpha T_s)}{\alpha^2}&T_s&0\end{bmatrix}, \tag{34}$$

where, $T_s = 1$ is the time step.

Note the complexity of both models, at first sight it might be beneficial to use the continuous dynamics model due to its simple structure, Q is diagonal, $\operatorname{trace}(\mathbf{A}) = 0$, noise is in two state variables only...

The measurement equation is the same for both models. The measurement function h is a discrete terrain map[5] represented by a table function that assigns vertical position (i.e. altitude) to each combination of latitude and longitude it covers. The measurement z_k is a terrain altitude below the vehicle which can be based on a barometric altimeter.

The noise v_k is distributed according to Gaussian mixture PDF with two components (this is simulating terrain with unmapped bridge or tunnel) [20]

$$p(v_k) = \sum_{g=1}^{2}\frac{1}{2}\mathcal{N}\{v_k;\hat{v}_g,P_g\}, \tag{35}$$

where the particular means and covariance matrices are given as follows $\hat{v}_1 = 0, \hat{v}_2 = 20$, $P_1 = P_2 = 1$.

The results can be seen in Table I for 50 Monte-Carlo simulations for three filters

- Efficient discrete point-mass filter with $N_{\text{pa}} = 34$ [1]–[3],
- Discrete bootstrap particle filter with 10^6 particles [21],

[5]The map is from Shuttle Radar Topography Mission (SRTM) an international project spearheaded by the U.S. National Geospatial-Intelligence Agency (NGA) and the U.S. National Aeronautics and Space Administration (NASA), see https://www2.jpl.nasa.gov/srtm/index.html.

	RMSE$^{(1)}$	RMSE$^{(2)}$	RMSE$^{(3)}$	RMSE$^{(4)}$	ASTD$^{(1)}$	ASTD$^{(2)}$	ASTD$^{(3)}$	ASTD$^{(4)}$	TIME
Efficient discrete	14.6867	13.2812	10.071	6.9791	18.2505	14.989	12.606	8.6158	0.82499
PF bootstrap	19.0446	17.9669	11.9933	9.2139	29.7935	29.1996	16.862	14.1169	0.46089
Efficient spectral	14.4769	13.0677	9.8878	6.8588	18.8222	15.0611	13.1594	8.807	0.41649

Table I: Results for state estimation in tracking scenario.

- Designed SD-based continuous point-mass filter with $N_{\mathrm{pa}} = 34$ (Algorithm in Section V.B).

The performance of the filters is compared using

- RMSE$^{(j)}$ =

$$\sqrt{\frac{1}{M(T+1)}\sum_{m=1}^{M}\sum_{k=0}^{T}((\mathbf{x}_{t_k}^{(j)})^{[m]} - (\hat{\mathbf{x}}_{t_k|t_k}^{(j)})^{[m]})^2}, \tag{36}$$

- ASTD$^{(j)}$ =

$$\sqrt{\frac{1}{M(T+1)}\sum_{m=1}^{M}\sum_{k=0}^{T}(\mathbf{P}_{t_k|t_k}^{(j,j)})^{[m]}}, \tag{37}$$

using $M = 50$ Monte-Carlo (MC) simulations, where $(\mathbf{x}_{t_k}^{(j)})^{[m]}$ is $(j+1)$-th element for the true state at time k and m-th MC simulation, $(\hat{\mathbf{x}}_{t_k|t_k}^{(j)})^{[m]} = \mathsf{E}[\mathbf{x}_{t_k}^{[i]}|\mathbf{z}^{t_k}]$ its filtering estimate, and $(\mathbf{P}_{t_k|t_k}^{(j,j)})^{[m]} = \mathsf{var}[(\mathbf{x}_{t_k}^{(j)})^{[m]}|\mathbf{z}^{t_k}]$ the corresponding filtering covariance diagonal element.

The table indicates, that the proposed filter provides most accurate and consistent results with the lowest computational complexity. The higher computational complexity of the discrete efficient filter is due to a need for two fast Fourier transform (FFT) transforms and one inverse FFT (IFFT) as opposed to the efficient spectral continuous filter which needs just one FFT and one IFFT each step. Also, in the discrete case where the FFT is used to perform efficient convolution using the convolution theorem, the FFT is performed on larger tensors (in general), because of the need for zero padding [22, pp. 27].

Based on the results, it may be beneficial to use continuous filter as it can be simple, efficient, and accurate.

VII. Concluding Remarks

The paper dealt with state estimation of continuous in time model with discrete measurement. In particular, the spectral differentiation has been introduced for the efficient solution to the Fokker-Planck equation describing the time evolution of the predictive PDF. This method preserves the computational complexity of efficient formulation of point mass filter $\mathcal{O}(N \log N)$ while achieving superior space convergence rate of $\mathcal{O}(c^N)(O < c < 1)$.

It was shown that the continuous spectral based estimation is less computationally complex and can be defined and implemented in a straightforward way similarly to the discrete grid and particle based state estimation. Despite simple and efficient implementation, the spectral approach leads to better estimation performance. Moreover, an inherent advantage of the continuous state-space model is its simplicity compared to the discrete counterpart especially in tracking and navigation areas.

In the future research we will focus on more accurate time stepping for spectral methods (i.e., an alternative to Euler method) and on alternative spectral approaches.

VIII. Acknowledgment

J. Duník, and J. Matoušek's work was co-funded by the European Union under the project ROBOPROX - Robotics and advanced industrial production (reg. no. CZ.02.01.01/ 00/22_008/0004590).

References

[1] J. Matouek, J. Dunk, M. Brandner, C. G. Park, and Y. Choe, "Efficient point mass predictor for continuous and discrete models with linear dynamics," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5937–5942, 2023, 22nd IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896323009928

[2] J. Duník, J. Matoušek, and O. Straka, "Design of efficient point-mass filter for linear and nonlinear dynamic models," *IEEE Control Systems Letters*, vol. 7, pp. 2005–2010, 2023.

[3] J. Matoušek, J. Duník, and M. Brandner, "Design of efficient point-mass filter with illustration in terrain aided navigation," in *26th International Conference on Information Fusion (FUSION)*, Charleston, USA, 2023, 2023.

[4] S. I. Denisov, W. Horsthemke, and P. Hänggi, "Generalized fokker-planck equation: Derivation and exact solutions," *The European Physical Journal B*, vol. 68, no. 4, pp. 567–575, 2009. [Online]. Available: https://doi.org/10.1140/epjb/e2009-00126-3

[5] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[6] F. Daum, "Nonlinear filters: beyond the kalman filter," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.

[7] H. Risken and H. Haken, *The Fokker-Planck Equation: Methods of Solution and Applications Second Edition*. Springer, 1989.

[8] B. Ng, A. Pfeffer, and R. Dearden, "Continuous time particle filtering," pp. 1360–1365, 2005.

[9] J. Maljaars, "When Euler meets Lagrange - particle-mesh modeling of advection dominated flows," Ph.D. dissertation, Technische Universiteit Delft, Nederland, 12 2019.

[10] S. Li and L. Petzold, "Moving mesh methods with upwinding schemes for time-dependent PDEs," *Journal of Computational Physics*, vol. 131, no. 2, pp. 368–377, 1997.

[11] K. Anonsen and O. Hallingstad, "Terrain aided underwater navigation using point mass and particle filters," in *2006 IEEE/ION Position, Location, And Navigation Symposium*, 2006, pp. 1027–1035.

[12] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*. Society for Industrial and Applied Mathematics, 2007. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9780898717839

[13] D. K. Salkuyeh, "Positive integer powers of the tridiagonal Toeplitz matrices," in *International Mathematical Forum*, vol. 22, 2006, pp. 1061–1065.

[14] G. Strang, *Computational Science and Engineering*. Wellesley, MA: Wellesley-Cambridge Press, Nov. 2007.

[15] L. N. Trefethen, *Spectral methods in MATLAB*. USA: Society for Industrial and Applied Mathematics, 2000.

[16] S. G. Johnson, "Notes on FFT-based differentiation," 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:1087476

[17] D. Durdiev and D. Durdiev, "The Fourier spectral method for determining a heat capacity coefficient in a parabolic equation," *Turkish Journal of Mathematics*, vol. 46, no. 8, pp. 3223–3233, 2022.

[18] H. Uecker, "A short ad hoc introduction to spectral methods for parabolic pde and the navier-stokes equations," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:201822478

[19] L. X. R. and V. P. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[20] P.-J. Nordlund and F. Gustafsson, "Marginalized particle filter for accurate and reliable terrain-aided navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 4, pp. 1385–1399, 2009.

[21] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, 2001, ch. An Introduction to Sequential Monte Carlo Methods, (Ed. Doucet A., de Freitas N., and Gordon N.).

[22] K. Iftekharuddin and A. Awwal, *Field Guide to Image Processing*. SPIE, 03 2012.