Learning Low-Rank Tensor Representations of Transition Densities for Prediction

Daniel Frisch[*], **Joshua Gehlen**[†], and **Uwe D. Hanebeck**[*]
[*]Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Karlsruhe Institute of Technology (KIT), Germany,
daniel.frisch@kit.edu, uwe.hanebeck@kit.edu
[†]Fraunhofer FKIE, Wachtberg, Germany, joshua.gehlen@fkie.fraunhofer.de

***Abstract*—We propose a simple strategy to perform the prediction step of grid-based Bayesian state estimation using the Candecomp/Parafac (CP) tensor decomposition. The rationale is that the Chapman-Kolmogorov equation, involving density multiplication and marginalization, reduces to univariate inner products once both the state density and the transition density (TD) are available in CP format. Thus, we can perform highly efficient prediction with cost linear in the state dimension, provided we have a low-rank CP representation of the TD. We propose several methods to learn such a representation from the system equations by minimizing a weighted L_2 distance: closed-form Gaussian factor functions for affine dynamics, and discretized loading vectors for block-diagonal models such as constant velocity and constant acceleration, where the fitting cost is exponential only in the block dimension and is accelerated further via the multidimensional FFT. For time-invariant systems, the decomposition is computed offline once and reused in every online prediction step, whose rank is bounded by the rank of the TD. A numerical evaluation on a four-dimensional planar constant-velocity model with closed-form ground truth demonstrates the accuracy, memory, and run-time trade-offs.**

***Index Terms*—Bayesian state estimation, nonlinear filtering, prediction, Chapman-Kolmogorov equation, grid-based filter, point-mass filter, curse of dimensionality, low-rank tensor decomposition, Candecomp/Parafac, transition density, constant velocity model, FFT.**

I. Introduction

Bayesian state estimation for discrete-time stochastic systems alternates between two steps: the *prediction step* propagates the state density through the system model, and the *filtering step* incorporates the current measurement [1]. This paper is concerned with the prediction step, which is governed by the Chapman-Kolmogorov equation

$$f_{k+1}(\underline{x}_{k+1}) = \int_{\mathbb{R}^S} f_k^{\mathrm{T}}(\underline{x}_{k+1} \mid \underline{x}_k)\, f_k(\underline{x}_k)\, \mathrm{d}\underline{x}_k \ , \qquad (1)$$

i.e., the product of the current state density f_k with the Transition Density (TD) f_k^{T}, followed by marginalization over the current state $\underline{x}_k$. Closed-form solutions exist only in special cases, most prominently for linear systems with additive Gaussian noise, where (1) reduces to the prediction step of the Kalman filter [2]. In the general nonlinear, non-Gaussian case, we need approximations.

A. Nonlinear Filtering

Parametric filters restrict the state density to a fixed family [3]: the extended Kalman filter linearizes the system model, the unscented Kalman filter propagates deterministically chosen sigma points [4], the Smart Sampling Kalman filter increases accuracy with more deterministically chosen samples [5]. Particle filters represent the density by random samples [6], [7]; they handle arbitrary nonlinearities but suffer from sample degeneracy, and their results fluctuate between runs. Gaussian particle filters simplify the resampling, and progressive variants help to avoid the degeneracy [8]. The flexibility can be increased with Gaussian mixtures (GMs) [9], [10]. Prediction in this case works especially well if the TD is available as a GM as well [11].

Grid-based methods, also known as point-mass filters (PMFs), discretize part of the state space and propagate the density values on the grid points deterministically [12]. They are global, robust, and reproducible, and some effort has been devoted to efficient grid design [13] and to Fourier-based density representations [14], [15]. Their fundamental limitation is the curse of dimensionality: the number of grid points N_{grid} grows exponentially with the state dimension S, and a naive discretization of (1) even costs $O(N_{\mathrm{grid}}^2)$ operations per prediction step. For linear dynamics, the prediction step can be rewritten as a convolution and evaluated via the fast Fourier transform (FFT) in $O(N_{\mathrm{grid}} \log N_{\mathrm{grid}})$ operations [16], and Lagrangian formulations translate the grid along the dynamics instead of propagating densities across a static grid [17]. These techniques reduce the cost per grid point, but not the exponential number of grid points itself. For continuous-time dynamics, grid-based prediction means solving the Fokker–Planck equation (FPE) numerically, which is challenging already in moderate dimensions [18], [19], [20].

B. Low-Rank Representations

Low-rank tensor decompositions are the established tool for approximating high-dimensional arrays. For example, the CANDECOMP/PARAFAC (CP) decomposition [21], [22], [23] writes a tensor as a sum of R rank-one terms, while the Tensor Train (TT) decomposition [24] uses a chain of matrix products; see [25], [26] for comprehensive overviews. With functions and operators in this representation, the complexity scales linearly instead of exponentially in the dimension [27]. For instance, TT surrogates of general multivariate densities can be built from few density evaluations via cross approximation, after which marginals and samples are obtained at a cost linear in the dimension [28]. Such surrogates, like ours, require a bounded

and known region of interest. Densities can also be estimated from samples directly in TT format [29].

The FPE has been solved in low-rank tensor formats [30], [31], [32], [33], [34], e.g., to perform multi-target tracking [35], [36]. In discrete-time PMFs, both the grid density and the transition probability tensor have been approximated in TT format via black-box cross approximation, and the prediction has been carried out as a tensor contraction in TT format [37]. As the cross approximation does not guarantee nonnegativity, negative density values caused numerical instabilities there; this issue and the relation between correlation and TT ranks have been analyzed in [38]. By exploiting block-diagonal dynamics, as we do in Section IV-C, low-rank densities have been propagated in real time with a Lagrangian time update for terrain-aided navigation [39]. In our own prior work, we performed FPE prediction on a cylindrical [40] and a spherical [41] manifold in tensor decomposition format and studied the interpolation of grid densities given in tensor decomposition representation [42].

From a dynamical systems perspective, the TD is the kernel of the Perron–Frobenius operator, which propagates densities, and its discretization with indicator functions on a partition of the state space corresponds to Ulam's method or a Markov state model [43]. Low-rank tensor approximations of such transfer operators have been proposed to mitigate the curse of dimensionality, with the caveat that strong coupling between variables leads to high ranks [43]. In particular, Galerkin approximations of Koopman and Perron–Frobenius operators have been computed in TT format from simulation data to identify metastable and coherent sets [44]. These works are data-driven and aim at the spectral analysis of the operator, whereas we approximate the TD from known system equations in order to propagate densities.

While the grid-oriented approaches above keep the state density in a low-rank format, the prediction is still computed anew in every step, e.g., by time-stepping the FPE in TT format [30] or by a Lagrangian grid update [39], which is rather expensive or restricted to invertible linear dynamics. The complementary question of how to obtain, offline and once, a low-rank representation of the *Transition Density* itself, such that the online prediction reduces to a cheap, format-native operation consisting of inner products, has received little attention so far. In [37], the transition tensor is sampled entrywise on a grid that follows the predicted density and is hence re-approximated in every time step, whereas we fit it once to the continuous TD on a fixed grid, in closed form or blockwise where possible.

C. Key Idea

We define a distance measure between the true, desired transition density and an approximating CP model. We then learn an optimal low-rank model by minimizing this distance measure. We state a few distinct cases where this approach is particularly effective.

II. Notation

A. Rectilinear Grid

We use an axis-aligned rectilinear grid whose sampling intervals may differ from one dimension to another [45]. The grid-point mapping in an S-dimensional space is defined as $\underline{x}^{\mathrm{g}}\colon \times_{s=1}^{S} \{1, 2, \ldots, N_s\} \to \mathbb{R}^S$

$$\underline{x}^{\mathrm{g}}[\underline{n}] = \underline{x}^{\mathrm{g}}[n_1, \ldots, n_S] = \begin{bmatrix} x_1^{\mathrm{g}}[n_1] & \cdots & x_S^{\mathrm{g}}[n_S] \end{bmatrix}^\top \ , \quad (2)$$

where $\underline{n}$ is a multi-index with one component n_s for each dimension s, and $n_s \in \{1, \ldots, N_s\}$. Here, N_s denotes the number of nodes along dimension s, so the total number of grid points is $N_{\mathrm{grid}} = \prod_{s=1}^{S} N_s$. In the experiments in this paper, we use uniform spacing per dimension.

B. Transition Densities

We consider the Markovian system model

$$\underline{x}_{k+1} = g(\underline{x}_k) + \underline{w}_k \ , \quad (3)$$

with process noise density f_w. The corresponding TD is

$$f_k^{\mathrm{T}}(\underline{x}_{k+1} \mid \underline{x}_k) = f_w\big(\underline{x}_{k+1} - g(\underline{x}_k)\big) \ , \quad (4)$$

and satisfies

$$\int_{\mathbb{R}^S} f_k^{\mathrm{T}}(\underline{x}_{k+1} \mid \underline{x}_k)\, \mathrm{d}\underline{x}_{k+1} = 1 \quad \forall \underline{x}_k \in \mathbb{R}^S \ . \quad (5)$$

On the rectilinear grid, we use the discrete TD tensor

$$\mathbf{T}[\underline{n}_{k+1}, \underline{n}_k] \approx f_k^{\mathrm{T}}(\underline{x}^{\mathrm{g}}[\underline{n}_{k+1}] \mid \underline{x}^{\mathrm{g}}[\underline{n}_k])\, \Delta V \ , \quad (6)$$

with cell volume $\Delta V = \prod_{s=1}^{S} \Delta x_s$. For each fixed current-state index $\underline{n}_k$, the column sum should satisfy $\sum_{\underline{n}_{k+1}} \mathbf{T}[\underline{n}_{k+1}, \underline{n}_k] = 1$.

III. Brute Force Method

In small dimensions, we can simply compute and store the full transition tensor and compress it via CP decomposition.

For example, with a univariate model $x_{k+1} = g(x_k) + w_k$, the transition tensor is a matrix. Starting from the widely used benchmark dynamics [46], [6]

$$x_{k+1} = \frac{x_k}{2} + \frac{25 x_k}{1 + x_k^2} + w_k \ , \quad (7)$$

with bimodal noise

$$w_k \sim 0.6 \cdot \mathcal{N}(x; -2, 1) + 0.4 \cdot \mathcal{N}(x; 4, 5) \ , \quad (8)$$

we obtain the TD

$$f(x_{k+1} \mid x_k) = f_w\left(x_{k+1} - \frac{x_k}{2} - \frac{25 x_k}{1 + x_k^2}\right) \quad (9)$$

that we discretize on a univariate grid $x^{\mathrm{g}}[i]$, $i = 1, \ldots, N$, as

$$\mathbf{T}[i, j] = f(x^{\mathrm{g}}[i] \mid x^{\mathrm{g}}[j])\, \Delta x \ . \quad (10)$$

Then we apply Singular Value Decomposition (SVD),

$$\mathbf{T} = \mathbf{U}\, \mathbf{\Sigma}\, \mathbf{V}^\top, \qquad \mathbf{T}_R = \sum_{r=1}^{R} \sigma_r\, \underline{u}_r\, \underline{v}_r^\top \ , \quad (11)$$

which is the rank-R CP form in 2D. Figure 1 shows $\mathbf{T}$ for $N = 241$ nodes on $[-25, 25]$, its rank-8 reconstruction, and the relative error over the rank.

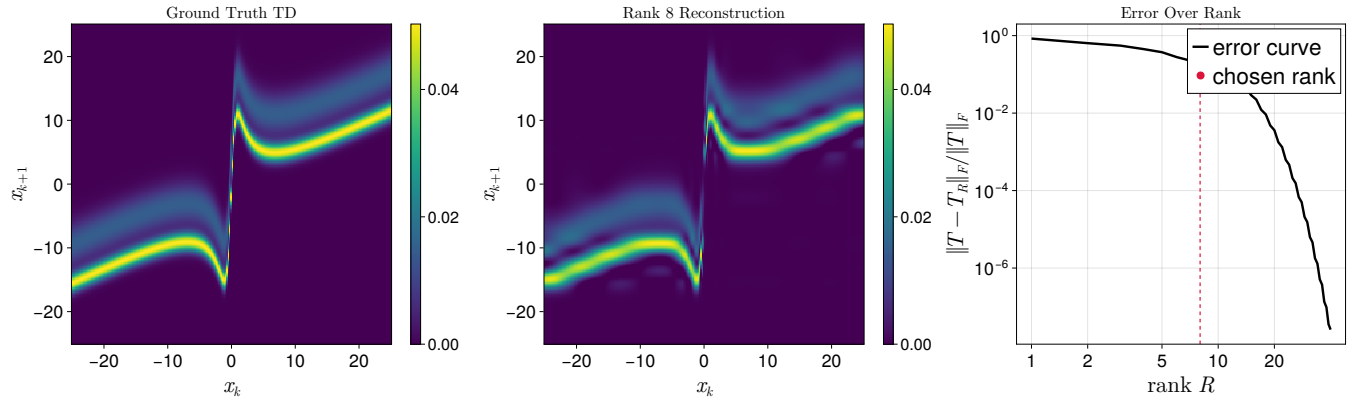


Fig. 1: Brute-force Kitagawa TD. Left: discretized ground-truth matrix $\mathbf{T}$ (column-normalized). Middle: low-rank reconstruction (column-normalized). Right: relative Frobenius error over rank, with the chosen rank highlighted.

IV. Parameter Learning

Sometimes the deviation between the CP representation of the desired TD and the ground truth TD can be written in closed form and computed without iterating the full dense grid. In that case, we can find the loading matrices via nonlinear optimization.

A. General Case

We seek a continuous rank-R CP representation of the TD with separable, per-dimension factor functions

$$f_R(\underline{x}_{k+1} \mid \underline{x}_k) = \sum_{r=1}^{R} \prod_{s=1}^{S} a_{r,s}(x_{k+1,s})\, b_{r,s}(x_{k,s}) \ , \qquad (12)$$

where $x_{k,s}$ denotes the s-th component of $\underline{x}_k$. The parameters of the factor functions are found by minimizing the weighted squared L_2 deviation from the true TD,

$$\begin{aligned} J &= \iint \left(f_k^{\mathrm{T}}(\underline{x}_{k+1} \mid \underline{x}_k) - f_R(\underline{x}_{k+1} \mid \underline{x}_k)\right)^2 \omega(\underline{x}_k)\, \mathrm{d}\underline{x}_{k+1}\ \mathrm{d}\underline{x}_k \\ &= \left\|f_k^{\mathrm{T}}\right\|_\omega^2 - 2\left\langle f_k^{\mathrm{T}},\, f_R\right\rangle_\omega + \|f_R\|_\omega^2 \ . \end{aligned} \qquad (13)$$

The weight $\omega(\underline{x}_k) = \prod_s \omega_s(x_{k,s}) \geq 0$ confines the fit to the region of interest. It is introduced because the TD has infinite energy, e.g., $\left\|f_k^{\mathrm{T}}\right\|_2^2 = \|f_w\|_2^2 \int \mathrm{d}\underline{x}_k = \infty$ for additive noise. Usually, the required rank grows with the extent of the region, see Section IV-E. With additive noise, the first term equals $\|f_w\|_2^2 \int \omega\, \mathrm{d}\underline{x}_k$, which is finite and independent of the parameters, so it can be discarded during optimization. The third term factorizes for any choice of factor functions into $2R^2S$ univariate integrals,

$$\|f_R\|_\omega^2 = \sum_{r_1=1}^{R}\sum_{r_2=1}^{R}\prod_{s=1}^{S} \int a_{r_1,s}\, a_{r_2,s}\, \mathrm{d}x \int b_{r_1,s}\, b_{r_2,s}\, \omega_s\, \mathrm{d}x \ . \qquad (14)$$

The critical part is the cross term

$$\left\langle f_k^{\mathrm{T}},\, f_R\right\rangle_\omega = \sum_{r=1}^{R} \iint f_w\big(\underline{x}_{k+1} - g(\underline{x}_k)\big) \cdot \prod_{s=1}^{S} a_{r,s}(x_{k+1,s})\, b_{r,s}(x_{k,s})\, \omega_s(x_{k,s})\, \mathrm{d}\underline{x}_{k+1}\ \mathrm{d}\underline{x}_k \ , \qquad (15)$$

a $2S$-dimensional integral whose numerical quadrature is exponentially expensive in S for general nonlinear g.

B. Gaussian Factor Functions + Affine Dynamics

The integral collapses to closed form for affine system models

$$g(\underline{x}_k) = \mathbf{G}\,\underline{x}_k + \underline{u} \ , \qquad (16)$$

Gaussian noise $f_w(\underline{w}) = \mathcal{N}(\underline{w};\, \underline{0},\, \mathbf{\Sigma}_w)$, and factor functions comprising a single Gaussian each,

$$a_{r,s}(x) = \alpha_{r,s}\, \mathcal{N}\big(x;\, \mu_{r,s}^{\mathrm{a}},\, \sigma_{r,s}^{\mathrm{a},2}\big) \ , \qquad (17)$$

and $b_{r,s}$ analogously with parameters $\beta_{r,s}, \mu_{r,s}^{\mathrm{b}}, \sigma_{r,s}^{\mathrm{b}}$. We choose a Gaussian weight $\omega_s(x) = \mathcal{N}(x;\, c_s,\, \gamma_s^2)$ covering the region of interest. As the product of two Gaussians, $b_{r,s}\,\omega_s$ is again an unnormalized Gaussian, so we absorb ω into the $b_{r,s}$ in the cross term. Then, the factor products in (15) are S-variate Gaussians with stacked means $\underline{\mu}_r^{\mathrm{a}}$, $\underline{\mu}_r^{\mathrm{b}}$, diagonal covariances $\mathbf{\Sigma}_r^{\mathrm{a}}$, $\mathbf{\Sigma}_r^{\mathrm{b}}$, and amplitudes $\bar{\alpha}_r = \prod_s \alpha_{r,s}$, $\bar{\beta}_r = \prod_s \beta_{r,s}$, and the Gaussian marginalization identity yields

$$\left\langle f_k^{\mathrm{T}},\, f_R\right\rangle_\omega = \sum_{r=1}^{R} \bar{\alpha}_r \bar{\beta}_r\, \mathcal{N}\big(\underline{\mu}_r^{\mathrm{a}} - \mathbf{G}\,\underline{\mu}_r^{\mathrm{b}};\, \underline{u},\, \mathbf{\Sigma}_w + \mathbf{\Sigma}_r^{\mathrm{a}} + \mathbf{G}\,\mathbf{\Sigma}_r^{\mathrm{b}}\mathbf{G}^\top\big) \ . \qquad (18)$$

Similarly, the univariate integrals in (14) evaluate to

$$\int a_{r,s}\, a_{s,s}\, \mathrm{d}x = \alpha_{r,s}\alpha_{s,s}\, \mathcal{N}\big(\mu_{r,s}^{\mathrm{a}};\, \mu_{s,s}^{\mathrm{a}},\, \sigma_{r,s}^{\mathrm{a},2} + \sigma_{s,s}^{\mathrm{a},2}\big) \ , \qquad (19)$$

and the b integrals in (14), which contain ω_s, are integrals over the product of three Gaussians and likewise available in closed form. Each summand in (18) costs $O(S^3)$ operations,

so one evaluation of J requires $O(R\,S^3 + R^2 S)$ operations, i.e., polynomial instead of exponential in S. Notably, $\mathbf{G}$ and $\mathbf{\Sigma}_w$ need not be diagonal, i.e., the TD itself need not factorize over the dimensions. Also, Gaussian mixture noise densities, as well as a mixture of affine models, merely introduce an additional sum over the respective components. All terms are differentiable in the means, variances, and amplitudes, enabling gradient-based optimization.

After the optimization, the factor functions can be discretized on the rectilinear grid to obtain the loading vectors of the TD tensor in CP form. The discrete representation can then be compressed further, e.g., via Alternating Least Squares (ALS), because its loading vectors are no longer restricted to Gaussian shape.

C. Block-Diagonal System Models

Alternatively, the factor functions can be arbitrary functions discretized on the rectilinear grid, i.e., piecewise constant on the grid cells with loading vectors $\underline{a}_{r,s}, \underline{b}_{r,s} \in \mathbb{R}^{N_s}$ as cell values. Here, ω is the indicator function of the grid domain, so the region of interest is the grid itself. Evaluating all integrals in (13) with the midpoint rule turns them into sums over grid points, i.e., J becomes the squared Frobenius norm of the difference between the discretized TD and its CP representation, where constant cell-volume factors are absorbed into the loading vectors. The Gram term (14) then factorizes into per-dimension inner products,

$$\|f_R\|_\omega^2 = \sum_{r=1}^{R}\sum_{s=1}^{R}\prod_{s=1}^{S}\left(\underline{a}_{r,s}^\top \underline{a}_{s,s}\right)\left(\underline{b}_{r,s}^\top \underline{b}_{s,s}\right) \;, \tag{20}$$

which is always cheap. The cross term (15), however, requires evaluating f_w at all N_{grid}^2 pairs of grid points, which is exponentially expensive in S. Tractability is retained whenever the TD factorizes into low-dimensional blocks. This is the case for block-diagonal $\mathbf{G} = \operatorname{blkdiag}(\mathbf{G}_1, \ldots, \mathbf{G}_P)$, and analogously block-diagonal $\mathbf{\Sigma}_w$, as in constant velocity (CV) and constant acceleration (CA) models, where each axis contributes an independent position–velocity(–acceleration) block

$$\mathbf{G}_p^{\text{CV}} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{G}_p^{\text{CA}} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \tag{21}$$

with sampling time T. The TD then factorizes over the blocks,

$$f_k^{\text{T}}(\underline{x}_{k+1} \mid \underline{x}_k) = \prod_{p=1}^{P} \mathcal{N}\left(\underline{x}_{k+1}^{(p)} - \mathbf{G}_p\,\underline{x}_k^{(p)};\, \underline{u}_p,\, \mathbf{\Sigma}_p\right) \;, \tag{22}$$

where $\underline{x}^{(p)}$ and $\underline{u}_p$ collect the dimensions of block p. Grouping the per-dimension loading vectors accordingly via Kronecker products,

$$\underline{a}_{r,p} = \bigotimes_{s \in \text{block } p} \underline{a}_{r,s} \in \mathbb{R}^{N_p} \;, \quad N_p = \prod_{s \in \text{block } p} N_s \;, \tag{23}$$

the cross term (15) becomes a product of per-block bilinear forms,

$$\left\langle f_k^{\text{T}},\, f_R \right\rangle_\omega = \sum_{r=1}^{R}\prod_{p=1}^{P} \underline{a}_{r,p}^\top \mathbf{T}_p\, \underline{b}_{r,p} \;, \tag{24}$$

with the block transition matrices

$$\mathbf{T}_p[\underline{n}, \underline{m}] = \mathcal{N}\left(\underline{x}_p^{\text{g}}[\underline{n}] - \mathbf{G}_p\,\underline{x}_p^{\text{g}}[\underline{m}];\, \underline{u}_p,\, \mathbf{\Sigma}_p\right) \Delta V_p \;, \tag{25}$$

where $\underline{n}$, $\underline{m}$ enumerate the N_p grid points $\underline{x}_p^{\text{g}}$ of block p and ΔV_p is the block cell volume. Note that the per-dimension CP representation still stores only the small loading vectors $\underline{a}_{r,s}$; the larger $\underline{a}_{r,p}$ and $\mathbf{T}_p$ are formed only for evaluating the objective. Each block matrix $\mathbf{T}_p$ comprises N_p^2 entries instead of N_{grid}^2.

Each bilinear form in (24) costs at most $O(N_p^2)$ operations, so evaluating the objective is exponential only in the block size and linear in the number of blocks P.

In general, the loading vectors are found iteratively, e.g., via ALS [25] or gradient-based optimization, because a CP decomposition of order three or higher has no closed-form solution. If, however, the loading vectors of a block are not required to factorize over its dimensions, i.e., each block is treated as a single mode, the TD can be approximated blockwise. Each $\mathbf{T}_p$ is then a matrix whose truncated SVD $\mathbf{T}_p \approx \mathbf{A}_p \mathbf{B}_p^\top$ with $\mathbf{A}_p, \mathbf{B}_p \in \mathbb{R}^{N_p \times R}$ is its best rank-R approximation in the Frobenius norm [47], as in Section III. The TD then takes the Kronecker form $\mathbf{T} \approx \bigotimes_p \mathbf{A}_p \mathbf{B}_p^\top$, which would comprise R^P terms if written as a single sum as in (24). However, it is applied blockwise, so a prediction of a density with rank R_f costs $O(PRR_f N_p)$, and R denotes the rank per block. A weight ω on $\underline{x}_k$ can be included by scaling the columns of $\mathbf{T}_p$ with $\sqrt{\omega}$ before and the right singular vectors with $1/\sqrt{\omega}$ after the decomposition; the result is then optimal in the ω-weighted Frobenius norm. The dense $\mathbf{T}_p$ with N_p^2 entries has to be formed for this, unless an iterative truncated SVD based on the matrix–vector products of Section IV-D is employed.

D. Multidimensional FFT Acceleration

The blockwise evaluation can be accelerated further whenever the block dynamics map grid points onto grid points. To make this precise, let D_p denote the number of dimensions in block p and write its grid points as $\underline{x}_p^{\text{g}}[\underline{n}] = \underline{x}_p^0 + \mathbf{\Delta}_p\,\underline{n}$, where $\underline{x}_p^0 \in \mathbb{R}^{D_p}$ collects the grid origins and diagonal matrix $\mathbf{\Delta}_p \in \mathbb{R}^{D_p \times D_p}$ contains the grid spacings Δx_s for $s \in \text{block } p$. The argument of the Gaussian in (25) then becomes

$$\begin{aligned} &\underline{x}_p^{\text{g}}[\underline{n}] - \mathbf{G}_p\,\underline{x}_p^{\text{g}}[\underline{m}] \\ &\quad = \underline{x}_p^0 + \mathbf{\Delta}_p\,\underline{n} - \mathbf{G}_p\left(\underline{x}_p^0 + \mathbf{\Delta}_p\,\underline{m}\right) \\ &\quad = \mathbf{\Delta}_p\left(\underline{n} - \mathbf{\Delta}_p^{-1}\,\mathbf{G}_p\,\mathbf{\Delta}_p\,\underline{m}\right) + (\mathbf{I} - \mathbf{G}_p)\,\underline{x}_p^0 \\ &\quad = \mathbf{\Delta}_p\left(\underline{n} - \mathbf{M}_p\,\underline{m}\right) + (\mathbf{I} - \mathbf{G}_p)\,\underline{x}_p^0 \;, \end{aligned} \tag{26}$$

with $\mathbf{M}_p = \mathbf{\Delta}_p^{-1}\,\mathbf{G}_p\,\mathbf{\Delta}_p$.

If $\mathbf{M}_p$ has integer entries, $\mathbf{T}_p[\underline{n}, \underline{m}]$ depends on the indices only through the generalized index difference $\underline{d} = \underline{n} - \mathbf{M}_p\underline{m}$,

$$\begin{aligned} \mathbf{T}_p[\underline{n}, \underline{m}] &= t_p[\underline{n} - \mathbf{M}_p\,\underline{m}] \ , \\ t_p[\underline{d}\,] &= \mathcal{N}\big(\boldsymbol{\Delta}_p\,\underline{d} + (\mathbf{I} - \mathbf{G}_p)\,\underline{x}_p^0;\ \underline{u}_p,\ \boldsymbol{\Sigma}_p\big)\,\Delta V_p \ . \end{aligned} \tag{27}$$

For example, a CV block with position spacing Δx and velocity spacing Δv, (26) yields $\mathbf{M}_p = \begin{bmatrix} 1 & T\Delta v/\Delta x \\ 0 & 1 \end{bmatrix}$, which is integer whenever $T\Delta v$ is an integer multiple of Δx; the natural choice $T\Delta v = \Delta x$ gives a unit shear. Like all $\mathbf{M}_p$ arising from CV and CA blocks, this matrix is unit upper triangular and hence unimodular, $\det \mathbf{M}_p = 1$, so its inverse is integer as well and $\underline{l} = \mathbf{M}_p\underline{m}$ is a bijection on $\mathbb{Z}^{D_p}$.

Substituting (27) into the bilinear forms of (24) gives

$$\begin{aligned} \underline{a}_{r,p}^{\top}\mathbf{T}_p\,\underline{b}_{r,p} &= \sum_{\underline{n}}\sum_{\underline{m}} a_{r,p}[\underline{n}]\ t_p[\underline{n} - \mathbf{M}_p\underline{m}]\ b_{r,p}[\underline{m}] \\ &= \sum_{\underline{d}} t_p[\underline{d}\,]\ c_{r,p}[\underline{d}\,] \ , \end{aligned} \tag{28}$$

with the generalized cross-correlation

$$c_{r,p}[\underline{d}\,] = \sum_{\underline{m}} a_{r,p}[\mathbf{M}_p\underline{m} + \underline{d}\,]\ b_{r,p}[\underline{m}] \ . \tag{29}$$

Because $\mathbf{M}_p$ is unimodular, we can substitute $\underline{l} = \mathbf{M}_p\underline{m}$ and pre-warp the current-state factor,

$$\tilde{b}_{r,p}[\underline{l}\,] = \begin{cases} b_{r,p}[\mathbf{M}_p^{-1}\underline{l}\,] & \text{if } \mathbf{M}_p^{-1}\underline{l} \text{ is a valid grid index,} \\ 0 & \text{otherwise,} \end{cases}$$

which is a pure index shuffle, e.g., for the unit-shear CV block, $\tilde{b}_{r,p}[l_1, l_2] = b_{r,p}[l_1 - l_2,\ l_2]$. Thereby, (29) turns into an ordinary D_p-dimensional cross-correlation

$$c_{r,p}[\underline{d}\,] = \sum_{\underline{l}} \tilde{b}_{r,p}[\underline{l}\,]\ a_{r,p}[\underline{l} + \underline{d}\,] \ . \tag{30}$$

The warped support fits into a bounding box with

$$\tilde{N}_{p,s} = \sum_{j} \big|[\mathbf{M}_p]_{sj}\big|\,(N_j - 1) + 1 \tag{31}$$

indices along dimension s. Zero-padding both $a_{r,p}$ and $\tilde{b}_{r,p}$ to a common box with $L_{p,s} \geq N_s + \tilde{N}_{p,s} - 1$ nodes per dimension prevents circular aliasing, and the cross-correlation theorem yields

$$c_{r,p} = \mathcal{F}^{-1}\big(\mathcal{F}(a_{r,p}) \odot \overline{\mathcal{F}(\tilde{b}_{r,p})}\big) \ , \tag{32}$$

where $\mathcal{F}$ denotes the D_p-dimensional discrete Fourier transform on the padded box, $\odot$ the elementwise or Hadamard product, and $\overline{(\cdot)}$ the complex conjugate. Using the FFT, each correlation costs $O(L_p \log L_p)$ operations with $L_p = \prod_s L_{p,s}$. Since the shear inflates the bounding box only by a constant factor, $L_p = O(N_p)$, so one evaluation of the cross term costs $O(R\,N_p \log N_p)$ per block instead of $O(R\,N_p^2)$.

Gradient-based optimization and ALS updates additionally require products of $\mathbf{T}_p$ with the loading vectors. Both directions are again convolutions,

$$\big(\mathbf{T}_p\,\underline{b}_{r,p}\big)[\underline{n}] = \sum_{\underline{l}} t_p[\underline{n} - \underline{l}\,]\ \tilde{b}_{r,p}[\underline{l}\,] \ , \tag{33}$$

$$\begin{aligned} \big(\mathbf{T}_p^{\top}\,\underline{a}_{r,p}\big)[\underline{m}] &= \gamma_{r,p}[\mathbf{M}_p\,\underline{m}] \ , \\ \gamma_{r,p}[\underline{l}\,] &= \sum_{\underline{n}} t_p[\underline{n} - \underline{l}\,]\ a_{r,p}[\underline{n}] \ , \end{aligned} \tag{34}$$

and are evaluated via (32) with the roles of the arguments exchanged accordingly. Hence, a complete ALS sweep costs $O(R\,L_p \log L_p)$ operations per block, and the memory footprint reduces from the N_p^2 entries of $\mathbf{T}_p$ to the $O(N_p)$ entries of the stencil t_p, which is never expanded into a full matrix.

E. Rank of the Transition Density

How well a TD compresses is decided by the process noise, not by the grid. In the CV case, the discrete TD is a smoothing kernel whose width is σ_s in position and σ_v in velocity, and the number of distinguishable kernel positions along an axis is proportional to the domain extent divided by that width. Consequently, the rank required for a fixed relative accuracy is

$$R \propto \frac{(x_{\max} - x_{\min})\,(v_{\max} - v_{\min})}{\sigma_s\,\sigma_v} = N_p\frac{\Delta x\,\Delta v}{\sigma_s\,\sigma_v} \ , \tag{35}$$

which is independent of the resolution.

This is favorable for our purpose: the finer the grid, the more the TD compresses relatively. The converse is equally important and limits the applicability of the method. A nearly deterministic TD, whose noise is narrower than one cell, approaches a permutation matrix, all of whose singular values are equal, and is therefore not compressible at all.

V. Prediction in CP Format

Let the initial state density be given with rank R_f,

$$f_k[\underline{n}_k] = \sum_{r_f=1}^{R_f}\prod_{s=1}^{S} g_{r_f,s}[n_{k,s}] \ , \tag{36}$$

and the TD by (12) on the same grid. Inserting both into the discrete counterpart of (1) separates the marginalization over the N_{grid} current-state points into univariate inner products,

$$\sum_{\underline{n}_k}\prod_{s=1}^{S} b_{r,s}[n_{k,s}]\,g_{r_f,s}[n_{k,s}] = \prod_{s=1}^{S}\underline{b}_{r,s}^{\top}\underline{g}_{r_f,s} \ , \tag{37}$$

so that

$$f_{k+1}[\underline{n}_{k+1}] = \sum_{r=1}^{R} c_r \prod_{s=1}^{S} a_{r,s}[n_{k+1,s}] \ , \tag{38}$$

$$c_r = \sum_{r_f=1}^{R_f}\prod_{s=1}^{S}\underline{b}_{r,s}^{\top}\,\underline{g}_{r_f,s} \ . \tag{39}$$

The prediction is again in CP form, its loading vectors are the predicted-state factors of the TD, i.e., the rank is R, the one

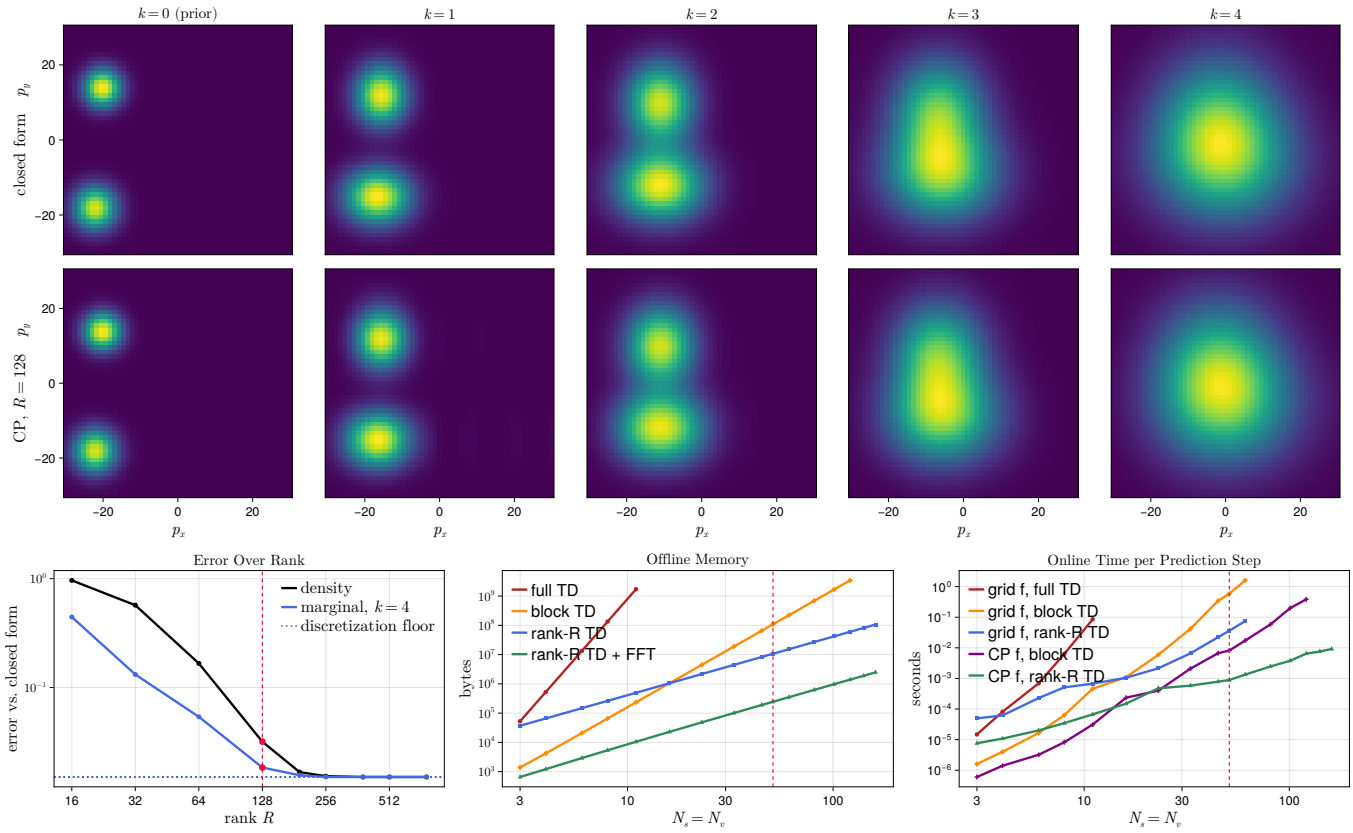


Fig. 2: Planar CV prediction with a Gaussian-mixture prior. Top: position marginal of the closed-form predicted density at $k = 0, \dots, 4$. Middle: the same marginal obtained with the rank-128 TD and the density in CP form on a grid with $N_s = N_v = 51$. Bottom left: relative error at $k = 4$ of the predicted density and of its position marginal over the rank, both measured against the closed form; the dotted line represents the discretization and truncation floor. Bottom center: measured memory of the stored TD required for offline optimization. Bottom right: time for one online prediction step for various combinations of a gridded or CP density f with a full, blockwise, or rank-128 blockwise TD. The vertical lines mark the choice used for visualization.

of the TD, and only the coefficients depend on f_k. One step costs $O(RR_fSN_s)$, i.e., it is linear in the dimension.

Neither nonnegativity nor unit mass of the TD is enforced during fitting; the factors are unconstrained and only fitted in the L_2 sense. Hence, the predicted density may contain small negative values, a known issue of low-rank density approximations [37], [38], and its mass may deviate from one, both on the order of the approximation error. In recursive prediction, these errors accumulate linearly, see the middle row of Fig. 2 for $k = 0, \dots, 4$. Combined with filtering steps, the overall deviation from a valid probability density may become significant over time. While the mass can easily be corrected, the overall method may have to be extended to enforce nonnegativity using an inherently nonnegative representation [48].

VI. Evaluation

We demonstrate the proposed method for a simple prediction where ground truth is available in closed form.

A. Setup

We consider a planar CV model with state $\underline{x}_k = \begin{bmatrix} p_x & v_x & p_y & v_y \end{bmatrix}^\top \in \mathbb{R}^4$, so that $\mathbf{G} = \mathrm{blkdiag}(\mathbf{G}_1^{\mathrm{CV}}, \mathbf{G}_2^{\mathrm{CV}})$ and $\mathbf{\Sigma}_w = \mathrm{blkdiag}(\mathbf{\Sigma}_1, \mathbf{\Sigma}_2)$ decompose into $P = 2$ identical position–velocity blocks with $\mathbf{\Sigma}_p = \mathrm{diag}(3.6^2, 2.0^2)$. Both axes of a block are discretized on $[-30, 30]$ with $N_s = N_v$ nodes, so the two spacings coincide, the sampling time is $T = \Delta x/\Delta v = 1$, and $\mathbf{M}_p = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$ is the unit shear of Section IV-D. A block carries $N_p = N_s N_v$ nodes and the planar grid $N_{\mathrm{grid}} = N_p^2$. The block matrices (25) are used without column renormalization, so that the gridded solution stays comparable to the continuous one. Each block is treated as a single mode and factorized via truncated SVD as described in Section IV-C, so R denotes the rank per block. Hence, the offline stage requires no iterative optimization in this experiment; ALS or gradient-based fitting would only be needed if the loading vectors had to factorize over position and velocity within a block.

The initial density is a two-component Gaussian mixture whose components are block-diagonal across the two spatial axes and carry position–velocity correlation. Under linear dynamics with Gaussian noise such a prior is propagated in closed form, per component and block, by $\underline{\mu} \leftarrow \mathbf{G}_p \underline{\mu}$ and $\mathbf{\Sigma} \leftarrow \mathbf{G}_p \mathbf{\Sigma} \mathbf{G}_p^\top + \mathbf{\Sigma}_p$. This provides a ground truth reference that is free of any discretization or truncation error.

B. Results

Figure 2 shows the planar position marginal at $N_s = N_v = 51$, i.e., $N_p = 2601$ per block, $N_{\text{grid}} \approx 6.8 \cdot 10^6$, and $\Delta x = \Delta v = 1.2$, over the first four prediction steps. The process noise spans 3.0 and 1.7 cells, so (35) predicts a compression of about $5\times$. The display rank is chosen to be $R = 128$, a compression of $20\times$, so that there is a slight deviation from the reference.

The error curves in the bottom left panel are measured after $k = 4$ steps against the continuous reference, once for the predicted density on the full four-dimensional grid and once for its position marginal. Both decrease with the rank until they reach their discretization and truncation floor.

The two right-hand panels show the offline memory and online computation cost. Offline, the panel reports the measured size of the stored TD. Materializing the full, dense TD already occupies 1.7 GB at $N_s = 11$. The P dense block matrices grow as N_p^2 and reach 3.4 GB at $N_s = 121$. The two low-rank representations proposed here instead grow only as N_p, where the version with FFT has a smaller constant.

For the online step, it does not matter whether the TD was obtained with or without FFT. The computational effort depends on how the density and the TD are each represented, and the rightmost panel compares five possible combinations. With the density in CP form, we have $O(PR_f N_p^2)$ using dense blocks and $O(PRR_f N_p)$ using factored blocks.

VII. Conclusion

We proposed a method to learn low-rank tensor representations of transition densities for certain time-invariant high-dimensional dynamical systems. This is done as an offline pre-computation. Thus, prediction is split into a slow offline and a fast online stage, enabling efficient prediction.

In particular, we can learn i) Gaussian factor functions from affine system dynamics and ii) general loading vectors from block-diagonal system matrices, where the offline training effort is exponential only in the dimension of the blocks. A numerical evaluation demonstrates a planar CV (4D state) prediction.

In the future, we plan to combine this with a Bayesian update step to form a recursive state estimator. As the rank of the proposed prediction is bounded by the rank of the employed TD representation, the proposed method could facilitate recursive state estimation without in-between iterative rank truncation.

References

[1] Simo Särkkä. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge: Cambridge University Press, 2013. DOI: doi.org/10.1017/CBO9781139344203.

[2] R. E. Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45. DOI: doi.org/10.1115/1.3662552.

[3] Kazufumi Ito and Kaiqi Xiong. "Gaussian Filters for Nonlinear Filtering Problems". In: *IEEE Transactions on Automatic Control* 45.5 (May 2000), pp. 910–927. DOI: doi.org/10.1109/9.855552.

[4] S. J. Julier and J. K. Uhlmann. "Unscented Filtering and Nonlinear Estimation". In: *Proceedings of the IEEE* 92.3 (2004), pp. 401–422. DOI: doi.org/10.1109/JPROC.2003.823141.

[5] Jannik Steinbring, Martin Pander, and Uwe D. Hanebeck. "The Smart Sampling Kalman Filter with Symmetric Samples". In: *Journal of Advances in Information Fusion* 11.1 (June 2016), pp. 71–90. URL: https://isif.org/media/smart-sampling-kalman-filter-symmetric-samples.

[6] Neil J. Gordon, David J. Salmond, and Adrian F. M. Smith. "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation". In: *IEE Proceedings F – Radar and Signal Processing* 140.2 (1993), pp. 107–113. DOI: doi.org/10.1049/ip-f-2.1993.0015.

[7] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering". In: *Statistics and Computing* 10.3 (July 2000), pp. 197–208. DOI: doi.org/10.1023/A:1008935410038.

[8] Uwe D. Hanebeck. "PGF 42: Progressive Gaussian Filtering with a Twist". In: *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*. Istanbul, Turkey, July 2013. URL: https://ieeexplore.ieee.org/document/6641119/.

[9] H. W. Sorenson and D. L. Alspach. "Recursive Bayesian Estimation Using Gaussian Sums". In: *Automatica* 7.4 (1971), pp. 465–479. DOI: doi.org/10.1016/0005-1098(71)90097-5.

[10] Daniel Frisch and Uwe D. Hanebeck. "Gaussian Mixture Particle Filter Step based on Method of Moments". In: *Proceedings of the 27th International Conference on Information Fusion (FUSION 2024)*. Venice, Italy, July 2024. DOI: doi.org/10.23919/FUSION59988.2024.10706458.

[11] Ondřej Straka and Uwe D. Hanebeck. "Efficient Gaussian Mixture Filters based on Transition Density Approximation". In: *Proceedings of the 28th International Conference on Information Fusion (FUSION 2025)*. Rio de Janeiro, Brazil, July 2025. DOI: doi.org/10.23919/FUSION65864.2025.11124060.

[12] Miroslav Simandl, Jakub Královec, and Torsten Söderström. "Advanced Point-Mass Method for Nonlinear State Estimation". In: *Automatica* 42.7 (2006), pp. 1133–1145. DOI: doi.org/10.1016/j.automatica.2006.03.010.

[13] Felipe Giraldo-Grueso, Andrey A. Popov, Uwe D. Hanebeck, and Renato Zanetti. "Optimal Grid Point Sampling for Point Mass Filtering". In: *Proceedings of the AAS/AIAA Space Flight Mechanics Meeting, Kaua'i, Hawaii*. Kaua'i, Hawaii, Jan. 2025.

[14] Florian Pfaff, Kailai Li, and Uwe D. Hanebeck. "Fourier Filters, Grid Filters, and the Fourier-Interpreted Grid Filter". In: *Proceedings of the 22nd International Conference on Information Fusion (Fusion 2019)*. Ottawa, Canada, July 2019. DOI: doi.org/10.23919/FUSION43075.2019.9011318.

[15] Florian Pfaff, Gerhard Kurz, and Uwe D. Hanebeck. "Multivariate Angular Filtering Using Fourier Series". In: *Journal of Advances in Information Fusion* 11.2 (Dec. 2016), pp. 206–226. URL: https://isif.org/media/multivariate-angular-filtering-using-fourier-series.

[16] Jakub Matoušek, Jindřich Duník, Marek Brandner, Chan Gook Park, and Yeongkwon Choe. "Efficient Point Mass Predictor for Continuous and Discrete Models with Linear Dynamics". In: *IFAC-PapersOnLine* 56.2 (2023). DOI: doi.org/10.1016/j.ifacol.2023.10.621.

[17] Jindřich Duník, Jan Krejčí, Jakub Matoušek, Marek Brandner, and Yeongkwon Choe. "Lagrangian Grid-based Estimation of Nonlinear Systems with Invertible Dynamics". In: *IFAC Journal of Systems and Control* (2026). DOI: doi.org/10.1016/j.ifacsc.2026.100385.

[18] Subhash Challa and Yaakov Bar-Shalom. "Nonlinear Filter Design Using Fokker-Planck-Kolmogorov Probability Density

Evolutions". In: *IEEE Transactions on Aerospace and Electronic Systems* 36.1 (2000), pp. 309–315. DOI: doi.org/10.1109/7.826335.

[19] Lukas Pichler, Arif Masud, and Lawrence A. Bergman. "Numerical Solution of the Fokker–Planck Equation by Finite Difference and Finite Element Methods—A Comparative Study". In: *Computational Methods in Stochastic Dynamics: Volume 2*. Ed. by Manolis Papadrakakis, George Stefanou, and Vissarion Papadopoulos. Dordrecht: Springer Netherlands, 2013, pp. 69–85. DOI: doi.org/10.1007/978-94-007-5134-7_5.

[20] Mrinal Kumar and Suman Chakravorty. "Nonlinear Filter Based on the Fokker-Planck Equation". In: *Journal of Guidance, Control, and Dynamics* 35.1 (2012), pp. 68–79. DOI: doi.org/10.2514/1.54070.

[21] Frank L. Hitchcock. "The Expression of a Tensor or a Polyadic as a Sum of Products". In: *Journal of Mathematics and Physics* 6.1-4 (1927), pp. 164–189. DOI: doi.org/10.1002/sapm192761164.

[22] Richard A. Harshman. "Foundations of the PARAFAC Procedure: Models and Conditions for an "Explanatory" Multimodal Factor Analysis". In: *UCLA Working Papers in Phonetics* 16 (1970), pp. 1–84.

[23] J. Douglas Carroll and Jih-Jie Chang. "Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of "Eckart–Young" Decomposition". In: *Psychometrika* 35.3 (1970), pp. 283–319. DOI: doi.org/10.1007/BF02310791.

[24] Ivan V. Oseledets. "Tensor-Train Decomposition". In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317. DOI: doi.org/10.1137/090752286.

[25] Tamara G. Kolda and Brett W. Bader. "Tensor Decompositions and Applications". In: *SIAM Review* 51.3 (2009), pp. 455–500. DOI: doi.org/10.1137/07070111X.

[26] Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. "Tensor Decomposition for Signal Processing and Machine Learning". In: *IEEE Transactions on Signal Processing* 65.13 (2017), pp. 3551–3582. DOI: doi.org/10.1109/TSP.2017.2690524.

[27] Gregory Beylkin and Martin J. Mohlenkamp. "Algorithms for Numerical Analysis in High Dimensions". In: *SIAM Journal on Scientific Computing* 26.6 (2005), pp. 2133–2159. DOI: doi.org/10.1137/040604959.

[28] Sergey Dolgov, Karim Anaya-Izquierdo, Colin Fox, and Robert Scheichl. "Approximation and sampling of multivariate probability distributions in the tensor train decomposition". In: *Statistics and Computing* 30.3 (May 2020), pp. 603–625. DOI: doi.org/10.1007/s11222-019-09910-z.

[29] Georgii S. Novikov, Maxim E. Panov, and Ivan V. Oseledets. "Tensor-Train Density Estimation". In: *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*. Ed. by Cassio de Campos and Marloes H. Maathuis. Vol. 161. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 1321–1331. URL: https://proceedings.mlr.press/v161/novikov21a.html.

[30] Sergey V. Dolgov, Boris N. Khoromskij, and Ivan V. Oseledets. "Fast Solution of Parabolic Problems in the Tensor Train/Quantized Tensor Train Format with Initial Application to the Fokker–Planck Equation". In: *SIAM Journal on Scientific Computing* 34.6 (2012), A3016–A3038. DOI: doi.org/10.1137/120864210.

[31] Andrei Chertkov and Ivan Oseledets. "Solution of the Fokker–Planck Equation by Cross Approximation Method in the Tensor Train Format". In: *Frontiers in Artificial Intelligence* 4 (2021), p. 668215. DOI: doi.org/10.3389/frai.2021.668215.

[32] Yifei Sun and Mrinal Kumar. "Nonlinear Bayesian Filtering Based on Fokker-Planck Equation and Tensor Decomposition". In: *2015 18th International Conference on Information Fusion (Fusion)*. 2015, pp. 1483–1488.

[33] Muhammad Altamash Khan, Martin Ulmke, Bruno Demissie, Felix Govaers, and Wolfgang Koch. "Combining Log-Homotopy Flow With Tensor Decomposition Based Solution for Fokker-Planck Equation". In: *2016 19th International Conference on Information Fusion (FUSION)*. 2016, pp. 2229–2236.

[34] Bruno Demissie, Muhammad Altamash Khan, and Felix Govaers. "Nonlinear Filter Design Using Fokker-Planck Propagator in Kronecker Tensor Format". In: *2016 19th International Conference on Information Fusion (FUSION)*. 2016, pp. 1–8.

[35] Felix Govaers, Bruno Demissie, Altamash Khan, Martin Ulmke, and Wolfgang Koch. "Tensor Decomposition-Based Multitarget Tracking in Cluttered Environments". In: *Journal of Advances in Information Fusion* 14.1 (2019), p. 86. URL: https://isif.org/media/tensor-decomposition-based-multitarget-tracking-cluttered-environments.

[36] Felix Govaers. "On Statistics Based Prediction of Decomposed 1–6 Tensor Probability Density Functions". In: *2023 IEEE Symposium Sensor Data Fusion and International Conference on Multisensor Fusion and Integration (SDF-MFI)*. 2023, pp. 1–6. DOI: doi.org/10.1109/SDF-MFI59545.2023.10361353.

[37] Jakub Matoušek, Marek Brandner, Jindřich Duník, and Ivo Punčochář. "Tensor Train Discrete Grid-Based Filters: Breaking the Curse of Dimensionality". In: *IFAC-PapersOnLine* 58.15 (2024). DOI: doi.org/10.1016/j.ifacol.2024.08.498.

[38] Jiří Ajgl and Ondřej Straka. "Aspects of Density Approximation by Tensor Trains". In: *Proceedings of the 28th International Conference on Information Fusion (FUSION 2025)*. Rio de Janeiro, Brazil, 2025. DOI: doi.org/10.23919/FUSION65864.2025.11124077.

[39] Jakub Matousek, Jan Krejci, Jindrich Dunik, and Renato Zanetti. *Tensor Decompositions for Online Grid-Based Terrain-Aided Navigation*. 2026. URL: https://arxiv.org/abs/2601.07728.

[40] Daniel Frisch and Uwe D. Hanebeck. "Fokker-Planck Prediction on the Cylindric Manifold using Tensor Decomposition of a Regular Grid". In: *17th Symposium Sensor Data Fusion: Trends, Solutions, Applications (SDF 2025)*. Bonn, Germany, Nov. 2025. DOI: doi.org/10.1109/SDF67080.2025.11330240.

[41] Daniel Frisch and Uwe D. Hanebeck. "Estimating Rotation Speed on the Sphere: Low-Rank Fokker–Planck Prediction on $\mathbb{S}^2 \times \mathbb{R}$". In: *Proceedings of the 2026 IEEE International Conference on Multisensor Fusion and Integration (MFI 2026)*. Pilsen, Czechia, Sept. 2026.

[42] Daniel Frisch and Uwe D. Hanebeck. "Interpolation of Probability Densities on a Grid in Tensor Decomposition Representation". In: *Proceedings of the 29th International Conference on Information Fusion (FUSION 2026)*. Trondheim, Norway, June 2026.

[43] Stefan Klus, Feliks Nüske, Péter Koltai, Hao Wu, Ioannis Kevrekidis, Christof Schütte, and Frank Noé. "Data-Driven Model Reduction and Transfer Operator Approximation". In: *Journal of Nonlinear Science* 28.3 (Jan. 2018), pp. 985–1010. DOI: doi.org/10.1007/s00332-017-9437-7.

[44] Feliks Nüske, Patrick Gelß, Stefan Klus, and Cecilia Clementi. "Tensor-based Computation of Metastable and Coherent Sets". In: *Physica D: Nonlinear Phenomena* 427 (2021), p. 133018. DOI: doi.org/10.1016/j.physd.2021.133018.

[45] Dana El-Rushaidat, Raine Yeh, and Xavier M. Tricoche. "Accurate Parallel Reconstruction of Unstructured Datasets on Rectilinear Grids". In: *Journal of Visualization* 24.4 (2021), pp. 787–806. DOI: doi.org/10.1007/s12650-020-00740-0.

[46] Genshiro Kitagawa. "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models". In: *Journal of Computational and Graphical Statistics* 5.1 (1996), pp. 1–25. DOI: doi.org/10.2307/1390750.

[47] Carl Eckart and Gale Young. "The approximation of one matrix by another of lower rank". In: *Psychometrika* 1.3 (Sept. 1936), pp. 211–218. DOI: doi.org/10.1007/BF02288367.

[48] Joshua Gehlen, Daniel Frisch, Felix Govaers, Uwe D. Hanebeck, and Wolfgang Koch. "On Constructing a Low-Rank Approximation of a Density". In: *18th Symposium Sensor Data Fusion: Trends, Solutions, Applications (SDF 2026)*. Bonn, Germany, Nov. 2026.