

Estimating Correlated Angles Using the Hypertoroidal Grid Filter

Florian Pfaff, Kailai Li, and Uwe D. Hanebeck

Abstract—Estimation for multiple correlated quantities generally requires considering a domain whose dimension is equal to the sum of the dimensions of the individual quantities. For multiple correlated angular quantities, considering a hypertoroidal manifold may be required. Based on a Cartesian product of d equidistant one-dimensional grids for the unit circle, a grid for the d -dimensional hypertorus can be constructed. This grid is used for a novel filter. For n grid points, the update step is in $O(n)$ for arbitrary likelihoods and the prediction step is in $O(n^2)$ for arbitrary transition densities. The run time of the latter can be reduced to $O(n \log n)$ for identity models with additive noise. In an evaluation scenario, the novel filter shows faster convergence than a particle filter for hypertoroidal domains and is on par with the recently proposed Fourier filters.

I. INTRODUCTION

Angular quantities arise in many real-world phenomena and may describe, e.g., wind directions, phases of signals, or orientations. It has been known for a long time that probability densities on periodic manifolds should consider the periodicity of the domain [1]. An example of a simple periodic manifold is the unit circle \mathbb{S}^1 , which can be parameterized by an angle in $[0, 2\pi)$. There are applications, such as in bioinformatics [2]–[5], that involve multiple correlated angles. A vector comprising d angles lies on a d -dimensional hypertorus \mathbb{T}^d , which is the Cartesian product of d circles. When keeping track of correlated angles, such as when receiving a signal with multiple antennas, continuously estimating wind directions at two stations close to each other, or tracking the orientations of soccer players, recursive estimators for hypertori are required.

A naïve approach would be to simply use a Kalman filter as on Euclidean domains. With some modifications [6], this approach can work for small uncertainties because the domain is locally similar to a Euclidean domain. However, the larger the uncertainties get, the less reliable this approach becomes. In a recently proposed filter for the two-dimensional torus [7], only the parameters of a parametric density are estimated, which makes it conceptually similar to the Kalman filter. Another approach that is more general and can be applied for arbitrary systems is the particle filter (PF) [8]. However, the PF only yields a set of particles and not a density that can be evaluated on the continuous domain.

Recently, two filters for hypertoroidal domains were presented [9], which are called the Fourier filters. In these filters,

densities and likelihoods are approximated using trigonometric polynomials (i.e., a Fourier series with finite nonzero coefficients). Two variants were proposed. In the Fourier identity filter (IFF), the density is directly approximated using a trigonometric polynomial. Due to approximation errors, the approximation of the density may have function values less than 0, which is an invalid value for density functions. In the second filter, the Fourier square root filter (SqFF), the trigonometric polynomial approximates the square root of the density. By squaring the function values of the trigonometric polynomial, a valid density function can be obtained. In [9], an update step for arbitrary likelihood functions in $O(n \log n)$ and a prediction step for identity system models with additive noise in $O(n)$ were presented. A general prediction step in $O(n^2)$ for the IFF and $O(n^2 \log n)$ for SqFF was provided in [10, Ch. 6].

In this paper, we present a novel grid-based filter for hypertori, which we call the hypertoroidal grid filter (HTGF). Grid filters are an established approach for bounded parts of Euclidean spaces [11]. While they can only cover a finite area, this is not a strong limitation for periodic domains because they are inherently of limited size. We provide an update step, a general prediction step, and a faster prediction step for identity system models with additive noise. The HTGF is related to grid filters for arbitrary likelihoods and system models on the sphere [12] and the hyperhemisphere [13]. For these topologies, no fast prediction step was presented. The HTGF is also related to the grid filter for the circle presented in [14] that can only be applied for identity system models with additive noise. We adopt the idea of [14] to use trigonometric polynomials as an interpolation scheme to provide continuous densities. To ensure the nonnegativity of the approximations, we use insights gained from the Fourier filters, which are closely related to our novel approach. An overview of the relationship between the HTGF and the Fourier filters is provided in the appendix.

The paper contains two key contributions. First, a grid filter for hypertoroidal manifolds is derived. Second, a grid generation scheme is presented that allows implementing a faster prediction step. The paper is structured as follows. In Sec. II, we describe how we represent the continuous density using grid values and how a continuous approximation can be derived from them. In the third section, we derive the HTGF. In Sec. V, we provide a conclusion and an outlook.

II. DENSITY REPRESENTATION AND APPROXIMATION

We start by describing how densities can be represented using a finite number of grid values in the first subsection.

Florian Pfaff, Kailai Li, and Uwe D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany {florian.pfaff, kailai.li, uwe.hanebeck}@kit.edu.

In the second subsection, we provide details on the grid generation scheme we use in our filter.

A. Grid Representations and Interpretations

In the derivation of our filter, we use a partition $\mathcal{A} = \{A_1, \dots, A_n\}$ of the domain \mathbb{T}^d that comprises n regions that do not overlap but cover the entire domain when joined. In this paper, we limit ourselves to partitions comprising equally sized regions. In a grid-based representation, our knowledge about the state is described by a vector of grid values $\underline{\gamma}$. Different interpretations of these grid values exist, which we also explain in [12], [13]. In the first interpretation, all elements of the partition are considered to be different discrete states and the vector $\underline{\gamma}$ consists of the probability masses of a probability mass function (pmf) on the discrete space. In the second, the grid values describe the function values of a continuous density at the grid points $\underline{\beta}_1, \dots, \underline{\beta}_n$.

In the first interpretation, the grid value γ_i describes the probability mass in the region A_i . To obtain the probability mass in A_i based on a continuous density function, one can integrate the density over A_i . The prediction and update steps are essentially those of a Wonham filter [15]. A filter adopting these ideas for the unit circle was proposed in [16]. The Wonham filter requires matrices for its prediction and update steps. Obtaining these from the system and measurement models defined on the continuous domain involves integrals, which can be prohibitively expensive for hypertoroidal domains.

Therefore, we focus on the second interpretation for the HTGF and say the grid values $\gamma_1, \dots, \gamma_n$ describe the function values on the grid points $\underline{\beta}_1, \dots, \underline{\beta}_n$. To provide a continuous pdf, additional assumptions about the underlying pdf are required. For an interpolation using a piece-wise constant function, we use a partition such that (for all i from 1 to n) A_i contains $\underline{\beta}_i$ and the interpolation yields γ_i everywhere in A_i . A different approach to providing a smooth interpolation is to use trigonometric polynomials. In some tests, the interpolations based on trigonometric polynomials were closer to the actual densities than the interpolations based on piece-wise constant functions. However, in this interpolation scheme, the individual grid values not only influence the function values in A_i but also in other regions. Considering this would make the derivation of the filter more complicated, and thus, the interpolation scheme with piece-wise constant functions is used for the derivation of our filter.

However, the interpolation based on trigonometric polynomials can still be used to interpolate the filter results as the grid values are always function values on the grid, and interpolating them differently can still yield valid densities. Unlike an interpolation based on piece-wise constant functions, an interpolation based on trigonometric polynomials can lead to negative function values. To obtain a nonnegative interpolation, we employ additional steps reminiscent of the SqFF. We first take the square root of the function values, interpolate them, and finally square the function values of the interpolation. Since all values are squared in the last step, the

Algorithm 1: Generate Default Grid

Input: Number of grid points per dimension m ,
dimension d

Output: Grid points $\underline{\beta}_1, \dots, \underline{\beta}_{m^d}$

$\mathcal{I} \leftarrow \{ \frac{2\pi}{m}, 2 \cdot \frac{2\pi}{m}, \dots, (m-1) \cdot \frac{2\pi}{m} \};$ // 1-D grid

$\mathcal{B} \leftarrow \text{dfoldCartesianProduct}(\mathcal{I}, d);$

/ Turn $\{ \{ \frac{2\pi}{m}, \dots, \frac{2\pi}{m} \}, \{ \frac{2\pi}{m}, \dots, 2 \cdot \frac{2\pi}{m} \}, \dots \}$ into
 $\{ [\frac{2\pi}{m}, \dots, \frac{2\pi}{m}]^\top, [\frac{2\pi}{m}, \dots, 2 \cdot \frac{2\pi}{m}]^\top, \dots \}$. */*

$\{ \underline{\beta}_1, \dots, \underline{\beta}_{m^d} \} \leftarrow \text{TupelsToVectors}(\mathcal{B});$

function values are guaranteed to be nonnegative. One only has to ensure that the grid values are always nonnegative, which is also required for valid interpolations using piece-wise constant functions.

Generally, the grid points should be chosen carefully to ensure that they represent the corresponding regions well. Using the center of each region as the grid point is thus a reasonable choice. To obtain estimates that are useful and easy to interpret, the regions should be small by some additional measure, e.g., the maximum distance between two points in the region. Good grids could be obtained by employing tessellation schemes such as Lloyd's algorithm [17]. In the next subsection, we provide details on an easy grid generation scheme that we use for the HTGF. While the corresponding partition does not minimize the maximal distance between two points in each region, the grid is fast to generate and lends itself well to a smooth interpolation via a trigonometric polynomial.

B. Subdivision of \mathbb{T}^d and Grid Generation

A grid for a Cartesian product of manifolds can be derived from the Cartesian product of grids for the individual manifolds. Since \mathbb{T}^d is the Cartesian product of d circles, we take the d fold Cartesian product of a grid for the circle. As explained in Algorithm 1, we then convert the tuples into vectors to obtain a grid on \mathbb{T}^d . The number of grid points n for \mathbb{T}^d is the d th power of the number of grid points m for the circle, i.e., $n = m^d$ and $m = \log_d(n)$. The equidistant grid for \mathbb{S}^1 with m points that we use as the basis for the Cartesian product is trivial to provide. Starting at 0, one places a grid point after each $2\pi/m$, with the last point being at $2\pi - 2\pi/m$. Due to the periodicity of the circle, the distance between the first and the last point is also $2\pi/m$.

All points on the d -dimensional grid can be described by an integer vector $\underline{k} \in (\mathbb{Z}/m\mathbb{Z})^d = \{0, \dots, m-1\}^d$ according to $[\frac{2\pi}{m}k_1, \frac{2\pi}{m}k_2, \dots, \frac{2\pi}{m}k_m]^\top = \frac{2\pi}{m}\underline{k}$. There are partitions (which are compatible with this grid) in which each region can be fully specified by two angles for each dimension. A simple partition comprises regions that are described by $[2\pi k_1/m, 2\pi(k_1+1)/m) \times \dots \times [2\pi k_d/m, 2\pi(k_d+1)/m)$. This partition is easy to comprehend without taking the periodicity of the domain into account. However, the grid points are in the corners and not in the centers of the regions. A partition that ensures that the grid points are in the centers is $[2\pi j_1/m - \pi/m, 2\pi j_1/m + \pi/m) \times \dots \times [2\pi j_d/m - \pi/m, 2\pi j_d/m + \pi/m)$. In this definition, the periodicity has

to be taken into account because values outside of $[0, 2\pi)$ can arise. It is a valid partition since none of the points is in two regions, even when considering the periodicity.

This type of grid is well suited for both interpolations we consider. For the interpolation that is constant in each region, we can determine the region in which the point lies very efficiently. Since the grid points are spaced equidistantly along each dimension, modulo arithmetic can be used to find the region in which a point lies. Thus, the interpolation using a piece-wise constant function can be evaluated in $O(d)$. For the interpolation via a trigonometric polynomial, we can arrange the vector of grid values γ in a d -dimensional $m \times m \cdots \times m$ tensor Λ that stores the function value at $\frac{2\pi}{m} \underline{k}$ in the entry with the vector-valued index \underline{k} (our indices start with zero). When applying the fast Fourier transform (FFT) [18] to this tensor, we obtain a tensor of Fourier coefficients. This also holds when including the square root for the nonnegative interpolation we mentioned before. The effort is in $O(n \log n)$ for the transformation and in $O(n)$ for each function evaluation because all n Fourier coefficients are required for calculating a single function value.

III. THE HYPERTOROIDAL GRID FILTER

We now describe the HTGF, starting with our approach to density approximation and normalization in the first subsection. The update step presented in the second subsection and the general prediction step in the third subsection are similar to those of other grid filters (see [12], [13]). In the fourth subsection, a more efficient prediction step for identity system models with additive noise is presented. This prediction step makes use of the structure of the grid. As mentioned in Sec. II-A, we focus on the interpolation that provides a piece-wise constant function in our derivation. Regardless of the interpolation scheme, the nonnegativity of the grid values always needs to be preserved throughout the prediction and update steps to ensure that the interpolation always describes a valid density function.

A. Density Approximation and Normalization

We start by describing how we approximate and normalize densities on the hypertorus. Afterward, we give details on joint and conditional densities, which will be required in the prediction step.

1) *Obtaining Normalized Approximations:* To obtain a grid-based representation, we generate a grid as described in Sec. II-B and evaluate the density on the grid points. Since the true probability mass in a region may not correspond to the function value at the grid point multiplied by the size of the region, the interpolation of the grid values may not be normalized.

To obtain a normalized density, we first calculate the integral over the unnormalized density \check{f} obtained by interpolating the grid values $\check{\gamma}$. For this, we use that the integral over the entire domain is equal to the sum of the integrals over all regions in the partition. Further, we use that all regions are

equally sized. This leads to the formula

$$\begin{aligned} \int_{\mathbb{T}^d} \check{f}(\underline{x}) d\underline{x} &= \sum_{i=1}^n \int_{A_i} \check{f}(\underline{x}) d\underline{x} = \sum_{i=1}^n \check{\gamma}_i \int_{A_i} 1 d\underline{x} \\ &= \frac{(2\pi)^d}{n} \sum_{i=1}^n \check{\gamma}_i = (2\pi)^d \text{mean}(\check{\gamma}). \end{aligned} \quad (1)$$

By dividing all the grid values by a constant, we can scale the function represented by the grid values. Thus, we can obtain the vector $\underline{\gamma}$ describing a normalized density from the vector describing an unnormalized one via $\underline{\gamma} = \check{\gamma}/(2\pi)^d \text{mean}(\check{\gamma})$. In the appendix of [14], we have proven that the interpolation based on trigonometric polynomials (both with and without the modification involving the square root) is also normalized in the one-dimensional case. Evidently, if we initialize our filter by deriving the grid values from the initial prior density as described in this subsection, we start with a vector containing only nonnegative values.

2) *Joint and Conditional Densities:* In the derivation of the general prediction step, we will use joint densities $f(\underline{x}_{t+1}, \underline{x}_t)$ and conditional densities $f(\underline{x}_{t+1}|\underline{x}_t)$. The joint density is a $2d$ -dimensional function, while the conditional density is a d -dimensional function for every fixed \underline{x}_t . The conditional density can also be interpreted as a $2d$ -dimensional function when the variable that it is conditioned on is not fixed and we allow both \underline{x}_{t+1} and \underline{x}_t to vary. Thus, we can use grid values for a grid on \mathbb{T}^{2d} to describe both conditional and joint densities.

For joint and conditional densities, we store the grid values, which correspond to the function values of $f(\beta_i, \beta_j)$ or $f(\beta_i|\beta_j)$, in a matrix Γ . The index i is increased along the columns and j along the rows. A valid joint density should integrate to 1 when integrating over both \underline{x}_{t+1} and \underline{x}_t . A conditional density $f(\underline{x}_{t+1}|\underline{x}_t)$ should integrate to 1 for every fixed \underline{x}_t . However, in our tests, normalizing the approximation of the conditional density did not offer a benefit over allowing unnormalized ones.

B. Update Step

We can perform an update step with any measurement model if we have the likelihood function $f_t^L(\underline{z}_t|\underline{x}_t)$ that describes the probability density of the measurement \underline{z}_t if the state is \underline{x}_t . For a specific measurement, $f_t^L(\underline{z}_t|\underline{x}_t)$ is a function on \mathbb{T}^d . Before the update step, we have a prior density $f_t^P(\underline{x}_t|\hat{\underline{z}}_1, \dots, \hat{\underline{z}}_{t-1})$ that considers all measurements from the initialization until time step $t-1$ (or none at all at $t=1$). Bayes' rule provides us a way to determine the posterior density $f_t^E(\underline{x}_t|\hat{\underline{z}}_1, \dots, \hat{\underline{z}}_t)$ that considers all measurements until time step t according to

$$\begin{aligned} f_t^E(\underline{x}_t|\hat{\underline{z}}_1, \dots, \hat{\underline{z}}_t) &= \frac{f_t^L(\hat{\underline{z}}_t|\underline{x}_t) f_t^P(\underline{x}_t|\hat{\underline{z}}_1, \dots, \hat{\underline{z}}_{t-1})}{\int_{\mathbb{T}^d} f_t^L(\hat{\underline{z}}_t|\underline{x}_t) f_t^P(\underline{x}_t|\hat{\underline{z}}_1, \dots, \hat{\underline{z}}_{t-1}) d\underline{x}_t} \\ &\propto \underbrace{f_t^L(\hat{\underline{z}}_t|\underline{x}_t) f_t^P(\underline{x}_t|\hat{\underline{z}}_1, \dots, \hat{\underline{z}}_{t-1})}_{\check{f}_t^E(\underline{x}_t|\hat{\underline{z}}_1, \dots, \hat{\underline{z}}_t)}. \end{aligned}$$

The update step can thus be seen as a multiplication of the prior density and the likelihood (which yields $\check{f}_t^E(\underline{x}_t|\hat{\underline{z}}_1, \dots, \hat{\underline{z}}_t)$) followed by a normalization.

Algorithm 2: Update Step of the HTGF

Input: Current grid values γ_t^p , grid points β_1, \dots, β_n , likelihood function f_t^L , measurement \hat{z}_t
Output: New grid values γ_t^e
 /* Determine unnormalized grid values. */
for $i \leftarrow 1$ **to** n **do**
 | $\check{\gamma}_{t,i}^e \leftarrow \gamma_{t,i}^p f_t^L(\hat{z}_t | \beta_i)$;
end
 $\gamma_t^e = \frac{n}{(2\pi)^d} \check{\gamma}_t^e / \sum_{i=1}^n (\check{\gamma}_{t,i}^e)$; // Normalize result.

In our filter, we have γ_t^p (which comprises the function values of $f_t^p(\mathbf{x}_t | \hat{z}_1, \dots, \hat{z}_{t-1})$) from the previous prediction step or the initialization. The function values of $\check{f}_t^e(\mathbf{x}_t | \hat{z}_1, \dots, \hat{z}_t)$ can then be obtained by multiplying the values in γ_t^p with the likelihoods at the grid points, i.e.,

$$\check{\gamma}_t^e = [\gamma_{t,1}^p f_t^L(\hat{z}_t | \beta_1) \quad \dots \quad \gamma_{t,n}^p f_t^L(\hat{z}_t | \beta_n)]^\top. \quad (2)$$

If γ_t^p contains the actual function values of f_t^p , $\check{\gamma}_t^e$ contains the true function values of the unnormalized posterior density. To obtain a vector of grid values γ_t^e describing a normalized density, we apply a normalization, as explained in Sec. III-A.1, to $\check{\gamma}_t^e$. All steps are summarized in Algorithm 2.

In general, γ_t^e does not contain the actual function values of $f_t^e(\mathbf{x}_t | \hat{z}_1, \dots, \hat{z}_t)$ because the normalization constant used may not be the same constant that would normalize $\check{f}_t^e(\mathbf{x}_t | \hat{z}_1, \dots, \hat{z}_t)$. For example, if the function values of the unnormalized posterior are high in regions where there are no grid points, the value of the integral over the true unnormalized posterior may be larger than $(2\pi)^d \text{mean}(\check{\gamma}_t^e)$.

C. General Prediction Step

The Chapman–Kolmogorov equation

$$f_{t+1}^p(\mathbf{x}_{t+1} | \hat{z}_1, \dots, \hat{z}_t) = \int_{\mathbb{T}^d} \underbrace{f_t^T(\mathbf{x}_{t+1} | \mathbf{x}_t) f_t^e(\mathbf{x}_t | \hat{z}_1, \dots, \hat{z}_t)}_{f_t^j(\mathbf{x}_{t+1}, \mathbf{x}_t | \hat{z}_1, \dots, \hat{z}_t)} d\mathbf{x}_t$$

provides a formula for the prior density of the next time step $f_{t+1}^p(\mathbf{x}_{t+1} | \hat{z}_1, \dots, \hat{z}_t)$ based on all measurements until time step t . We can implement it by performing a multiplication (which yields the joint density $f_t^j(\mathbf{x}_{t+1}, \mathbf{x}_t | \hat{z}_1, \dots, \hat{z}_t)$) and subsequently marginalizing the result. In this subsection, we begin by explaining how both operations can be realized based on the grid values. At the end of this subsection, the two operations are merged into one concise formula.

We start with the vector γ_t^e from the previous update step and the matrix Γ_t^T describing $f_t^T(\mathbf{x}_{t+1} | \mathbf{x}_t)$, which we generate as explained in Sec. III-A.2. For all $(i, j) \in \{1, \dots, n\}^2$, we obtain the function value of $f_t^j(\beta_i, \beta_j | \hat{z}_1, \dots, \hat{z}_t)$ via $\gamma_{t,[i,j]}^T \gamma_{t,j}^e$, with $\gamma_{t,[i,j]}^T$ describing the entry at row i and column j of Γ_t^T . Thus, we can obtain all entries of the matrix Γ_t^j , which describes $f_t^j(\mathbf{x}_{t+1}, \mathbf{x}_t | \hat{z}_1, \dots, \hat{z}_t)$. Then, we use formula (1) to marginalize \mathbf{x}_t out and obtain

$$\gamma_{t+1,i}^p = \frac{(2\pi)^d}{n} \sum_{j=1}^n \gamma_{t,[i,j]}^j = \frac{(2\pi)^d}{n} \sum_{j=1}^n \gamma_{t,[i,j]}^T \gamma_{t,j}^e. \quad (3)$$

Algorithm 3: General Prediction Step for Time-Variant Transition Densities

Input: Current grid values γ_t^e , transition density f_t^T , grid points β_1, \dots, β_n
Output: New grid values γ_{t+1}^p
 $\Gamma_t^T \leftarrow \text{EvalAtCartProductOfPoints}(f_t^T, \{\beta_1, \dots, \beta_n\})$;
 $\gamma_{t+1}^p \leftarrow \frac{(2\pi)^d}{n} \Gamma_t^T \gamma_t^e$; // Predict using formula (4).

Multiplying the row vector $\gamma_{t,[i,:]}^T$, which is the i th row of Γ_t^T , and the column vector γ_t^e yields the same result as the rightmost sum in (3). To obtain the entire vector γ_{t+1}^p , we can thus use the matrix–vector product

$$\gamma_{t+1}^p = \frac{(2\pi)^d}{n} \Gamma_t^T \gamma_t^e. \quad (4)$$

Since all values in Γ_t^T and γ_t^e are nonnegative, the values in γ_{t+1}^p are nonnegative and can thus be used to obtain an interpolation that is a valid density. An overview of the general prediction step is given in Algorithm 3.

D. Prediction Step for Identity Models With Additive Noise

When the state is only perturbed by additive noise, the random variable for the state at the next time step \mathbf{x}_{t+1} can be written based on the state \mathbf{x}_t at time step t and the additive noise \mathbf{w}_t according to $\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{w}_t \pmod{2\pi}$ with $\mathbf{w}_t \sim f^w(\mathbf{w}_t)$. Then, $f_t^T(\mathbf{x}_{t+1} | \mathbf{x}_t) = f_t^w(\mathbf{x}_{t+1} - \mathbf{x}_t)$. Thus, we obtain the simpler form

$$f_{t+1}^p(\mathbf{x}_{t+1} | \hat{z}_1, \dots, \hat{z}_t) = \int_{\mathbb{T}^d} \underbrace{f_t^w(\mathbf{x}_{t+1} - \mathbf{x}_t) f_t^e(\mathbf{x}_t | \hat{z}_1, \dots, \hat{z}_t)}_{f_t^j(\mathbf{x}_{t+1}, \mathbf{x}_t | \hat{z}_1, \dots, \hat{z}_t)} d\mathbf{x}_t$$

for the Chapman–Kolmogorov equation.

In our efficient update step for identity models with additive noise, we consider the specific shape of the grid described in Sec. II-B. All grid points can be written as $\frac{2\pi}{m} \underline{k}$ with $\underline{k} \in (\mathbb{Z}/m\mathbb{Z})^d$. For two grid points $\frac{2\pi}{m} \underline{l}$ and $\frac{2\pi}{m} \underline{k}$ (with both \underline{l} and \underline{k} in $(\mathbb{Z}/m\mathbb{Z})^d$), we obtain

$$f_t^j\left(\frac{2\pi}{m} \underline{l}, \frac{2\pi}{m} \underline{k}\right) = f_t^w\left(\frac{2\pi}{m} \underline{l} - \frac{2\pi}{m} \underline{k}\right) f_t^e\left(\frac{2\pi}{m} \underline{k} | \hat{z}_1, \dots, \hat{z}_t\right)$$

as the formula for the joint density. Due to the 2π -periodicity along each dimension,

$$f_t^w\left(\frac{2\pi}{m} \underline{l} - \frac{2\pi}{m} \underline{k}\right) = f_t^w\left(\frac{2\pi}{m} (\underline{l} - \underline{k} \pmod{m})\right)$$

holds. Since $(\underline{l} - \underline{k} \pmod{m}) \in (\mathbb{Z}/m\mathbb{Z})^d$, the point $\frac{2\pi}{m} (\underline{l} - \underline{k} \pmod{m})$ is one of the grid points of the d -dimensional grid that we use for f_t^e . If we have the function values of the noise density on this grid, we have all relevant values to determine the function values of the joint density on the grid.

As explained in Sec. II-B, we can also arrange the grid values in d -dimensional $m \times \dots \times m$ tensors. We now do this to provide a formula for the tensor for the predicted density Λ_{t+1}^p based on the tensors Λ_t^e and Λ_t^w for the posterior density and the noise density. We denote individual entries, which are indexed using a vector-valued index l , by $\lambda[l]$. In this representation, $\lambda[l] = \gamma_i$ for $\beta_i = \frac{2\pi}{m} \underline{l}$. We now perform the

Algorithm 4: Prediction Step for Identity Models
 With Time-Variant Additive Noise Terms

Input: Current grid values as a tensor Λ_t^e , noise density f_t^w , grid points $\underline{\beta}_1, \dots, \underline{\beta}_n$
Output: New grid values as a tensor Λ_{t+1}^p
 $\Lambda_t^w \leftarrow \text{ToTensor}(\text{EvalAtPoints}(f_t^w, \{\underline{\beta}_1, \dots, \underline{\beta}_n\}))$;
 $\Lambda_{t+1}^p \leftarrow \left(\frac{2\pi}{m}\right)^d \text{CycConv}(\Lambda_t^w, \Lambda_t^e)$;

marginalization via (1) with the sum running over all vector-valued indices $\underline{k} \in (\mathbb{Z}/m\mathbb{Z})^d$ instead of over the indices 1 to n . Then, we obtain

$$\begin{aligned} \lambda_{t+1}^p[\underline{l}] &= \left(\frac{2\pi}{m}\right)^d \sum_{\underline{k} \in (\mathbb{Z}/m\mathbb{Z})^d} f_t^j\left(\frac{2\pi}{m}\underline{l}, \frac{2\pi}{m}\underline{k}\right) \\ &= \left(\frac{2\pi}{m}\right)^d \sum_{\underline{k} \in (\mathbb{Z}/m\mathbb{Z})^d} f_t^w\left(\frac{2\pi}{m}(\underline{l} - \underline{k} \bmod m)\right) \\ &\quad \cdot f_t^e\left(\frac{2\pi}{m}\underline{k}|\hat{z}_1, \dots, \hat{z}_t\right) \\ &= \left(\frac{2\pi}{m}\right)^d \sum_{\underline{k} \in (\mathbb{Z}/m\mathbb{Z})^d} \lambda_t^w[\underline{l} - \underline{k} \bmod m] \lambda_t^e[\underline{k}]. \end{aligned}$$

Evidently, this formula yields nonnegative values if Λ_t^w and Λ_t^e only contain nonnegative values. Since $O(n)$ operations are required to determine an individual grid value $\lambda_{t+1}^p[\underline{l}]$, naïvely computing all n entries of Λ_{t+1}^p would involve $O(n^2)$ operations. However, using a scaled cyclic (also called circular) convolution [19, Sec. 2.2.1] of Λ_t^w and Λ_t^e , the tensor Λ_{t+1}^p can be determined more efficiently. For the cyclic convolution, implementations involving FFTs can be used to obtain the result in $O(n \log n)$ [19, Sec. 3.3]. Finally, we can reshape the tensor Λ_{t+1}^p into a vector γ_{t+1}^p for the next update step. However, if the general prediction step is never used, we can stay in the tensor-based representation and also use it in the update step by adapting formula (2) appropriately. In Algorithm 4, we summarize the operations of this prediction step when staying in the tensor-based representation.

IV. EVALUATION

Our evaluation involves a scenario with an identity measurement model and a nonlinear system model. We compare the HTGF with the PF, IFF, and SqFF with different numbers of parameters (i.e., grid points, particles, or coefficients). All filters are available in the GitHub repository of [20]. The prediction step of the IFF in libDirectional is implemented using operations in $O(n^2 \log n)$ because this allows the use of highly optimized routines, leading to superior run times in tests. In the first subsection, we describe the evaluation scenario. Then, we explain the evaluation metrics and conclude the section with a presentation and discussion of the evaluation results.

A. Scenario Description

We consider a scenario on the hypertorus \mathbb{T}^3 for ten time steps from the initialization in time step 1 to the end of the simulation in time step 10. The initial state \underline{x}_1 is distributed according to a hypertoroidal uniform distribution. The system model is applied nine times to transition from the first to the

tenth time step. One measurement is obtained in each of the ten time steps. As the basis for the nonlinear system model, we use the (scalar) function (see [21])

$$a_\rho(\alpha) = \pi \cdot \left(\sin\left(\frac{\text{sign}(\alpha - \pi) |\alpha - \pi|^\rho}{2} + 1\right) \right),$$

which is a continuous bijection on \mathbb{S}^1 . The system model for our scenario on the hypertorus is

$$\begin{aligned} \underline{x}_{t+1} &= \begin{bmatrix} a_4(\mathbf{x}_{t,1}) \\ a_5(\mathbf{x}_{t,2}) \\ a_6(\mathbf{x}_{t,3}) \end{bmatrix} + \underline{w}_t, \\ \underline{w}_t &\sim f_{\text{WN}}\left(\underline{w}_t; \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.8 & 0.8 & 0.5 \\ 0.8 & 1 & 0.6 \\ 0.5 & 0.6 & 0.5 \end{bmatrix}\right), \end{aligned}$$

which involves the density f_{WN} of a trivariate wrapped normal distribution. The general prediction step needs to be employed due to the nonlinearity of the system function. The measurement model is

$$\underline{z}_t = h(\underline{x}_t) + \underline{v}_t, \quad \underline{v}_t \sim f_{\text{WN}}\left(\underline{v}_t; \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1.9 & 0.5 & 1.4 \\ 0.5 & 0.9 & 0.5 \\ 1.4 & 0.5 & 1.2 \end{bmatrix}\right).$$

B. Evaluation Metric

We compare all filters regarding the errors and run times. To calculate the error, we determine the distance

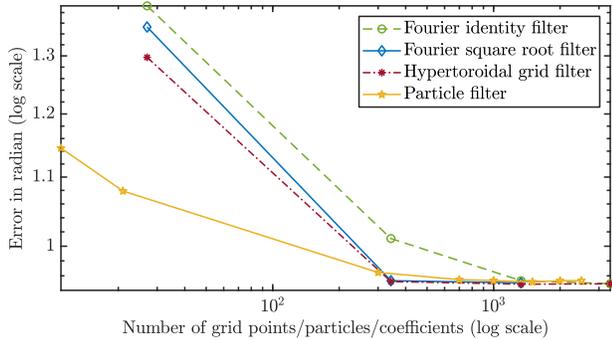
$$d_0(\hat{\underline{x}}_{10}, \tilde{\underline{x}}_{10}) = \sqrt{\sum_{i=1}^d \min(\hat{x}_{10,i} - \tilde{x}_{10,i}, 2\pi - (\hat{x}_{10,i} - \tilde{x}_{10,i}))^2}$$

between the estimate provided by the filter $\hat{\underline{x}}_{10}$ and the true value $\tilde{\underline{x}}_{10}$ at the last time step. The mean directions derived from the filters' states are used as the estimates. It should be noted that the mean direction does generally not minimize the error defined above (in the 1-D case, the mean direction minimizes an error involving the cosine, as shown in [22]).

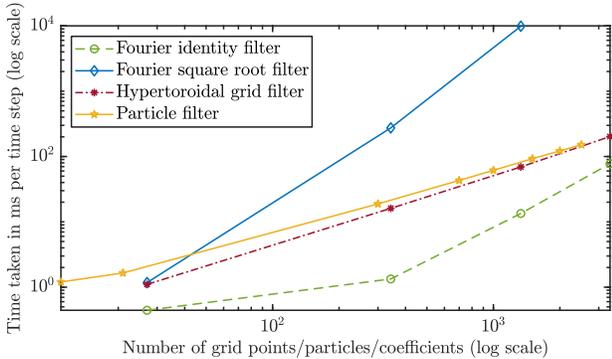
All run times were measured on a laptop with an Intel Core i7-7500U CPU and 16 GB of RAM running Matlab 2020a on Windows 10. The run times do not include computations involved for the initialization, such as drawing the initial samples for the PF, determining the matrix Γ^T , or generating the matrices for the Fourier filters. Since the accuracies of the filters vary from run to run, we averaged the errors and run times over 2500 runs.

C. Evaluation Results

The grid with the lowest number of grid points considered comprises 27 grid points and is based on the threefold Cartesian product of a grid on \mathbb{S}^1 with three grid points. As can be seen in Fig. 1a, a PF with 27 particles achieves a higher estimation accuracy. However, when using 5^3 parameters, the HTGF and the Fourier filters achieve an accuracy that is not even achieved by a PF with 2000 particles. Since the accuracies of the HTGF and the Fourier filters do not change significantly for higher numbers of parameters, we believe the mean direction determined by the filters is close to the mean direction of the true posterior density.



(a) Error over number of grid points, particles, or Fourier coefficients.



(b) Time over number of grid points, particles, or Fourier coefficients.

Fig. 1: Errors and run times for the evaluation scenario.

The increase in the run time of the PF is linear as the PF is in $O(n)$. However, in the PF, the noise density needs to be sampled for each particle in every prediction step, which is a rather expensive operation. While the HTGF has a complexity in $O(n^2)$, the operations it uses are cheaper and highly optimized routines are available in programming languages like Matlab. In the configurations considered, the IFF is faster than the HTGF, while the SqFF is slower (see Fig. 1b). The long run times of the SqFF are caused by the larger sizes of the tensors in its intermediate steps, which lead to huge increases in the computational effort in this scenario with a six-dimensional transition density.

When considering both the accuracy and run time, the PF is good for very few parameters but is then quickly outperformed by the HTGF and IFF for higher numbers of parameters. While the IFF is faster than the HTGF for equal numbers of parameters, the HTGF is more accurate. When comparing the IFF and HTGF for configurations of comparable run time, the accuracies of the mean directions are similar.

V. CONCLUSION AND OUTLOOK

Grid filters are versatile and can be applied with success to hypertoroidal manifolds. Grids for hypertoroidal domains can be generated based on the Cartesian product of grids for the circle. For the derivation of the update and prediction steps, we subdivided the domain into equally sized regions, each containing a grid point. While an interpolation that is constant in each region was used in the derivation of the formulae for the filter, a smooth interpolation can be provided based on

trigonometric polynomials. While the employed partitioning scheme is not optimal in regard to an additional criterion involving the maximum distance between two points in a region, it allows for a fast prediction step when the system is only perturbed by additive noise.

In our evaluation, the convergence of the mean direction of the filter to the true mean direction was much faster for the HTGF than for the PF. While the HTGF has a complexity in $O(n^2)$, it was slightly faster than the PF in the considered configurations. While the results may be different for scenarios that require even more parameters, the HTGF appears preferable in many scenarios due to its good convergence speed and deterministic nature. The HTGF was faster than the SqFF while having a comparable estimation accuracy, which indicates that the HTGF is the superior filter. In our evaluation, the IFF was faster but yielded less accurate results than the HTGF. When comparing configurations of comparable run times, the IFF and HTGF performed similarly. In such cases, we recommend the use of the HTGF as it always provides an interpolation that is guaranteed to be nonnegative on the entire domain. When choosing the optimal filter for a scenario with an identity system model with additive noise, one should consider that the HTGF has a complexity in $O(n)$ for the update step and $O(n \log n)$ for the prediction step. Since these complexities are swapped for the IFF, the HTGF may be faster than the IFF if multiple measurements are obtained in each time step.

Future work may involve considering other manifolds, such as discs and polydiscs. Further research on the Fourier filters may also lead to alternative ways to ensure the nonnegativity of the density represented by the coefficients. Last, the HTGF could be used as the basis to develop a filter for arbitrary Cartesian products of circles and Euclidean spaces via Rao–Blackwellization.

APPENDIX

Relationship of the HTGF to the Fourier Filters

In this appendix, we consider the relationship of the HTGF to the Fourier filters. The basic operations that are performed in the HTGF (multiplications, integrals, convolutions) are also done in the Fourier filters (albeit differently) in the frequency domain. While the interpolation of the HTGF is similar to that of the SqFF, the update and prediction steps are more closely related to the IFF. A thorough comparison of a grid filter and the Fourier filters is given for a related grid filter for the circle in [14]. In the evaluation in [14], the filters are compared based on the quality of the interpolations when performing an approximation, a prediction step, or an update step. However, the general prediction step is not considered.

a) Approximation: When no closed-form formulae are used to determine the Fourier coefficients, they are calculated based on the same function values on the grid that are used by the HTGF. For the SqFF, the square root of the function values is taken before performing the FFT. If we use the interpolation for the HTGF that involves taking the square root before obtaining the Fourier coefficients via an FFT, the functions provided by the SqFF and HTGF are identical.

b) *Update Step*: In the HTGF, each grid value is multiplied with the respective likelihood. Thus, we have a complexity in $O(n)$. The IFF and SqFF use a non-cyclic convolution of the Fourier coefficients of the likelihood and the prior density, leading to a complexity in $O(n \log n)$. If both the prior density and likelihood can be represented perfectly using n Fourier coefficients, the non-cyclic convolution yields the coefficients of the function that is the true multiplication result. However, the resulting coefficient vector may be longer than the original vectors because additional Fourier coefficients may be required to represent the multiplication result. To prevent an increase in the number of coefficients over time, the coefficient vector is truncated in the IFF and SqFF. This can be interpreted as low pass filtering the result. Besides the errors stemming from approximating the densities and likelihoods with trigonometric polynomials, this truncation is the only source of error in the Fourier filters.

The HTGF only uses n grid points to describe the multiplication result. Thus, we act as if the number of grid points was sufficient to precisely describe the multiplication result. If we disregard all errors from previous time steps, we obtain the true function values of the (unnormalized) product at the grid points. The interpolation of the grid values is generally not identical to the result of the SqFF. The characteristics of the error introduced by simply assuming the grid is sufficiently dense to precisely describe the multiplication result (although it is not) are different from those of the error introduced by a low pass filter.

Prediction Step for Identity Models with Additive Noise: In the grid filter, a cyclic convolution in $O(n \log n)$ is used. In the Fourier filters, an entry-wise product in $O(n)$ is used. Additional transformations are used in the SqFF. If the Fourier coefficients used in the IFF correspond to those obtained by applying FFTs to Λ_t^w and Λ_t^c , the prediction step of the HTGF and IFF are equivalent. The reason for this is that a cyclic convolution can be implemented using an FFT, an entrywise product, and an IFFT [19, Sec. 3.3].

c) *General Prediction Step*: The grid filter uses a matrix–vector product that is in $O(n^2)$. In the IFF, a convolution is required for determining all Fourier coefficients of the joint density. However, due to the marginalization that follows, only a part of the coefficients is required and all these entries can be obtained via a matrix–vector multiplication [10, Rem. 4] in $O(n^2)$. Under the assumption that the transition density and the posterior density are perfectly represented using the Fourier coefficients, no approximation errors are done in the IFF. The multiplication in the Chapman–Komogorov equation only increases the complexity of the function in \underline{x}_t and this dimension is marginalized out. In the HTGF, the formula for the joint density can already introduce an error, and thus, the marginalized result is generally not error-free. Therefore, there is no direct correspondence between the general prediction steps of the two filters.

ACKNOWLEDGMENT

This work is supported by the German Research Foundation (DFG) under grant HA 3789/16-1.

REFERENCES

- [1] W. Schmidt, “Statistische Methoden beim Gefügestudium krystalliner Schiefer,” *Sitzungsberichte der Kaiserlichen Akademie der Wissenschaften in Wien – mathematisch-naturwissenschaftliche Classe*, vol. 126, pp. 515–539, Jul. 1917.
- [2] K. V. Mardia, C. C. Taylor, and G. K. Subramaniam, “Protein Bioinformatics and Mixtures of Bivariate von Mises Distributions for Angular Data,” *Biometrics*, vol. 63, no. 2, pp. 505–512, 2007.
- [3] G. Hughes, “*Multivariate and Time Series Models for Circular Data with Applications to Protein Conformational Angles*,” Ph.D. thesis, The University of Leeds, The University of Leeds, p. 208, 2007.
- [4] W. Boomsma, K. V. Mardia, C. C. Taylor, J. Ferkinghoff-Borg, A. Krogh, and T. Hamelryck, “A Generative, Probabilistic Model of Local Protein Structure,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 26, pp. 8932–8937, 2008.
- [5] K. V. Mardia, G. Hughes, C. C. Taylor, and H. Singh, “A Multivariate von Mises Distribution with Applications to Bioinformatics,” *Canadian Journal of Statistics*, vol. 36, no. 1, pp. 99–109, 2008.
- [6] G. Kurz, I. Gilitschenski, and U. D. Hanebeck, “Recursive Bayesian Filtering in Circular State Spaces,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 31, no. 3, pp. 70–87, Mar. 2016.
- [7] G. Kurz, F. Pfaff, and U. D. Hanebeck, “Nonlinear Toroidal Filtering Based on Bivariate Wrapped Normal Distributions,” in *Proceedings of the 20th International Conference on Information Fusion (Fusion 2017)*, Xi’an, China, Jul. 2017.
- [8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [9] F. Pfaff, G. Kurz, and U. D. Hanebeck, “Multivariate Angular Filtering Using Fourier Series,” *Journal of Advances in Information Fusion*, vol. 11, no. 2, pp. 206–226, Dec. 2016.
- [10] F. Pfaff, “Multitarget Tracking Using Orientation Estimation for Optical Belt Sorting,” Ph.D. dissertation, Karlsruhe Institute of Technology, 2019, defended 2018.
- [11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents Series. The MIT Press, 2005.
- [12] F. Pfaff, K. Li, and U. D. Hanebeck, “The Spherical Grid Filter for Nonlinear Estimation on the Unit Sphere,” in *Proceedings of the 1st Virtual IFAC World Congress (IFAC-V 2020)*, Jul. 2020.
- [13] —, “A Hyperhemispherical Grid Filter for Orientation Estimation,” in *Proceedings of the 23rd International Conference on Information Fusion (Fusion 2020)*, Virtual, Jul. 2020.
- [14] —, “Fourier Filters, Grid Filters, and the Fourier-Interpreted Grid Filter,” in *Proceedings of the 22nd International Conference on Information Fusion (Fusion 2019)*, Ottawa, Canada, Jul. 2019.
- [15] W. M. Wonham, “Some Applications of Stochastic Differential Equations to Optimal Nonlinear Filtering,” *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, vol. 2, no. 3, pp. 347–369, 1964.
- [16] G. Kurz, F. Pfaff, and U. D. Hanebeck, “Discrete Recursive Bayesian Filtering on Intervals and the Unit Circle,” in *Proceedings of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, Baden-Baden, Germany, Sep. 2016.
- [17] S. Lloyd, “Least Squares Quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [18] J. W. Cooley and J. W. Tukey, “An Algorithm for the Machine Calculation of Complex Fourier Series,” *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.
- [19] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*. Prentice Hall, 1984.
- [20] G. Kurz, I. Gilitschenski, F. Pfaff, L. Drude, U. D. Hanebeck, R. Haeb-Umbach, and R. Y. Siegwart, “Directional Statistics and Filtering Using libDirectional,” *Journal of Statistical Software*, May 2019.
- [21] I. Gilitschenski, G. Kurz, and U. D. Hanebeck, “Non-Identity Measurement Models for Orientation Estimation Based on Directional Statistics,” in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D.C., USA, Jul. 2015.
- [22] R. S. Bucy, S. Y. Cheng, and A. J. Mallinckrodt, “A Design Study for an Optimal Non-Linear Receiver/Demodulator. Final Report,” 1970.