sensors

MDPI

Article

The State Space Subdivision Filter for Estimation on SE(2)

Florian Pfaff *, Kailai Li and Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany; kailai.li@kit.edu (K.L.); uwe.hanebeck@kit.edu (U.D.H.)
* Correspondence: pfaff@kit.edu; Tel.: +49-721-608-48831

Abstract: The $\mathrm{SE}(2)$ domain can be used to describe the position and orientation of objects in planar scenarios and is inherently nonlinear due to the periodicity of the angle. We present a novel filter that involves splitting up the joint density into a (marginalized) density for the periodic part and a conditional density for the linear part. We subdivide the state space along the periodic dimension and describe each part of the state space using the parameters of a Gaussian and a grid value, which is the function value of the marginalized density for the periodic part at the center of the respective area. By using the grid values as weighting factors for the Gaussians along the linear dimensions, we can approximate functions on the $\mathrm{SE}(2)$ domain with correlated position and orientation. Based on this representation, we interweave a grid filter with a Kalman filter to obtain a filter that can take different numbers of parameters and is in the same complexity class as a grid filter for circular domains. We thoroughly compared the filters with other state-of-the-art filters in a simulated tracking scenario. With only little run time, our filter outperformed an unscented Kalman filter for manifolds and a progressive filter based on dual quaternions. Our filter also yielded more accurate results than a particle filter using one million particles while being faster by over an order of magnitude.

Keywords: grid filter; nonlinear filtering; periodic manifold; special Euclidean group

check for updates

Citation: Pfaff, F.; Li, K.; Hanebeck, U.D. The State Space Subdivision Filter for Estimation on SE(2). *Sensors* **2021**, *21*, 6314. https://doi.org/10.3390/s21186314

Academic Editor: Andrey V. Savkin

Received: 18 August 2021
Accepted: 17 September 2021
Published: 21 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

When tracking vehicles or robots that are only capable of planar motion within rooms or on plains, their pose can be described by a two-dimensional position and a heading angle. Integrating the heading angle into the tracking is essential for realistic motion models of nonholonomic vehicles [1]. Such models may also include other angles, such as a trailer angle or the steering angle for slowly steering vehicles [2]. Estimating one's own position and orientation is also an essential component for simultaneous localization and mapping [3].

Each position–angle pair corresponds to an element in the special Euclidean group $\mathrm{SE}(2)$, which is the group of two-dimensional rigid-body motions (Section 10.6.2, [4]). Quantities in $\mathrm{SE}(2)$ are also relevant in other settings, such as in control [5]. However, it is nontrivial to rigorously estimate states in $\mathrm{SE}(2)$, which is why even nowadays, some trackers rely merely on estimating the velocities along the two axes [6]. No heading angle is required to use a classical constant velocity or constant acceleration model [7]. However, such simple models generally do not adequately take the kinematics of the vehicle or robot into account.

For estimation on $\mathrm{SE}(2)$, it is relevant to consider that the pose involves both an angle, which is a periodic quantity on the unit circle $\mathbb{S}$, and a real vector in $\mathbb{R}^2$. Stacking the orientation, described by an angle in $[0, 2\pi)$ and the position yields a three-dimensional vector, which is an element of the Cartesian product $\mathbb{S} \times \mathbb{R}^2$. While all entries are real valued, one needs to consider the periodicity of the angle and not treat the vector as a three-dimensional Euclidean vector.

One can provide a filter for $\mathrm{SE}(2)$ based on the extended Kalman filter [8]. However, such filters are generally suboptimal since they cannot fully take the periodicity of the angle

into account. Especially in cases with high uncertainties, the local linearization can cause a degradation of estimation accuracy. A very general filter that can be easily adapted to the $\mathrm{SE}(2)$ domain is the particle filter (PF) [9]. However, PFs suffer from particle degeneracy and impoverishment [10] and can be slow to converge [11]. Other general approaches that can be applied to $\mathrm{SE}(2)$ are the invariant extended and unscented Kalman filter [12] and the unscented Kalman filter on manifolds (UKF-M) [13]. An approach tailored explicitly to $\mathrm{SE}(2)$ that is based on dual quaternions was proposed in [14]. A filter based on sample reduction for scenarios in which only samples of the noise are available was presented in [15].

It is possible to make use of the property that the state can be described as a Cartesian product of a nonlinear domain and a linear domain. If we denote the part of our state describing the orientation with x^ω and the part describing its translation with $\underline{x}^\tau$, then we can rewrite our joint density as a product of the conditional density $f^c(\underline{x}^\tau | x^\omega)$ and the marginalized density for the periodic part $f^\omega(x^\omega)$. Splitting the state up into a part that is well suited for estimation using the Kalman filter and another part for which a nonlinear filter should be applied is a technique that has become popular for PFs as Rao–Blackwellization [16] and has also found application in other filters such as the point mass filter [17]. This technique was also the foundation for the grid-based filter for SE(2) [18].

The filter we present in this paper also builds upon the idea of splitting up the state. It differs from the filter proposed in [18] in several regards. First, system models in our novel filter should be given in the form of a transition density instead of an equation of random variables. Second, no way to provide a smooth continuous density was presented in [18]. In our current paper, we consider different ways to provide a continuous density based on the parameters of our filter.

Another limitation of [18] is that the conditionals for the likelihood and the system noise all have the same parameters. In our current paper, we have no such limitation. Thus, the system and measurement noise for the position can be orientation dependent. The system noise can depend on both the current angle and the potential future angle. Especially the dependency on the current angle is important because the uncertainty in the predicted position can depend on the orientation in many applications. For the likelihood, which describes the measurement model, orientation dependency is relevant since the orientation of an object can have an influence on the error in the position measurement. Last, unlike [18], we allow for system inputs that depend on the orientation, which can be the key to solving scenarios such as the evaluation scenario in this paper.

For the nonlinear part concerning the angle, we use a grid-based filter that was first presented in [19] and later extended for arbitrary nonlinear models and generalized to arbitrary-dimensional hypertori in [20]. The filter uses trigonometric polynomials [21] to provide an interpolation of the grid values that closely matches the original continuous density even when using small numbers of grid points. Similar filters were also presented for the (hyper)sphere and hyperhemisphere [22,23]. All these filters have in common that they can be interpreted to use a subdivision of the state space into non-overlapping regions. Since the domains in the grid filters mentioned in this paragraph are bounded, they can be subdivided into areas of finite size.

For our novel filter, which we call the state space subdivision filter (S3F), we subdivide the domain $\mathbb{S} \times \mathbb{R}^2$, which can be used to describe planar poses, into a finite number of non-overlapping parts. For this, we subdivide $\mathbb{S}$ into the set of areas $\mathcal{A} = \{A_1, \ldots, A_n\}$ with $n \in \mathbb{N}$. Each of the n areas in the subdivision is in the form $[a, b)$, with a and b in $[0, 2\pi)$. Note that for the unit circle, we get a valid interval even if $b < a$. In this case, the interval crosses the boundary of periodicity, which is located at 2π in our case. Based on the subdivision of $\mathbb{S}$, the subdivision of $\mathbb{S} \times \mathbb{R}^2$ is then $\{A_i \times \mathbb{R}^2 | i \in \{1, \ldots, n\}\}$. Please note that the focus of our explanations and derivations will be on the elements of $\mathcal{A}$ and that we will use the term area to refer to the subsets of the periodic domain and not subsets of the entire state space.

For the derivation of our new filter, we introduce three assumptions. For an angle α, the marginalized density $f^{\omega}(\alpha)$ and the conditional density $f^{c}(\underline{x}^{\tau}|\alpha)$ are assumed to only depend on which area $\mathcal{A}$ lies in and not on the precise value of α. Furthermore, we assume that $f^{c}(\underline{x}^{\tau}|\alpha)$ is a Gaussian density. This will allow us to use the basic ideas of the grid filter for the density describing the orientation and the Kalman filter for the conditional part. However, due to dependencies, significant adjustments are required.

While we focus on $\mathrm{SE}(2)$ in this paper, the S3F can be directly applied for $\mathbb{S} \times \mathbb{R}^d$ with $d \in \mathbb{N}$ and most key parts of our filter are not limited to $\mathbb{S} \times \mathbb{R}^d$. It can be adopted for other Cartesian products, such as of a hypertorus or hypersphere and a Euclidean domain, using grid filters for the respective domain [22,23]. To allow for easier visualization, we limit ourselves to plotting densities on $\mathbb{S} \times \mathbb{R}$. This domain could, e.g., be used to estimate the rotor angle and velocity of a motor. In Figure 1, we show the density of a partially wrapped normal distribution [24] on $\mathbb{S} \times \mathbb{R}$ with mean $[0, 0]^{\top}$ and a covariance matrix with 1 in the entries on the diagonal and 0.7 in the off-diagonal entries. An illustration of all the assumptions of our filter is given in Figure 2. We will later regard how better continuous densities can be derived from the parameters of the filter.

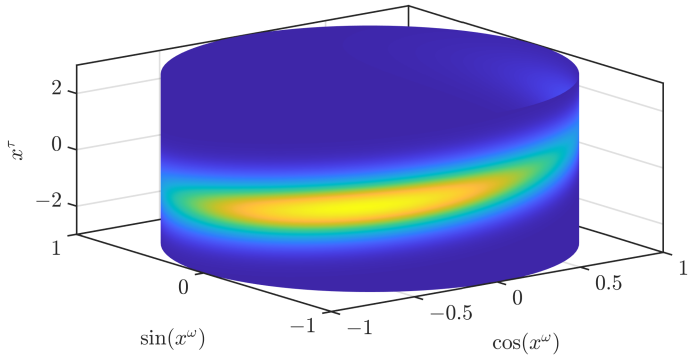

Figure 1. True density of the partially wrapped normal distribution.

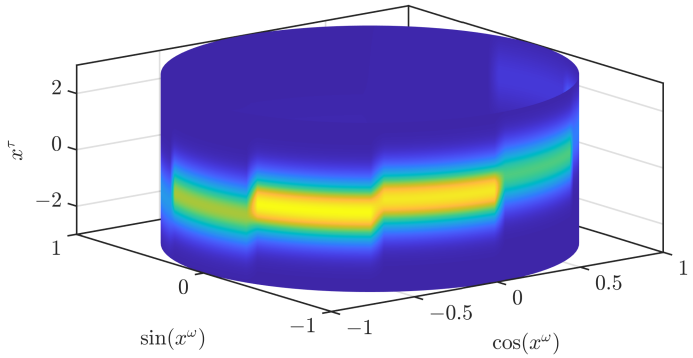

Figure 2. Naïve approximation of the density shown in Figure 1 based on a subdivision of the state space. 10 areas are used in this example.

The paper is structured as follows. In the next section, we go into detail on how we represent densities, how we approximate given densities, and how a continuous density can be provided. The update and prediction steps of the S3F are derived in Section 3. We evaluate the accuracy and run time of the S3F and other state-of-the-art filters in a simulated scenario in Section 4. A discussion, a conclusion, and an outlook are presented in Section 5.

2. Density Representation and Approximation

Since all areas in $\mathcal{A}$ are in the form of $[a, b)$, each area is (when considering the periodicity of the domain) connected and contains more than one element. For each area

A_i, we specify a grid point β_i in A_i, which is in some way representative of the area. For this, we define the topology-aware distance (Section 2.3.4, [25])

$$d_0(\alpha_1, \alpha_2) = \min(|\alpha_1 - \alpha_2|, 2\pi - |\alpha_1 - \alpha_2|) \tag{1}$$

for two angles α_1 and α_2.

We place the grid point β_i such that it is in A_i and has the same distance to a and b. To simplify the filter, we only consider subdivisions of the periodic domain into areas of equal size. Hence, for n regions, each area is of size $\frac{2\pi}{n}$. With all these specifications, the subdivision only depends on the starting point of the first region when $\mathcal{A}$ is always sorted such that the next region after $[a, b)$ is $[b, c)$. Without loss of generality, we say the first region is the region in which 0 lies. Thus, the starting point of the first region must lie in $(2\pi - \frac{2\pi}{n}, 0]$.

In our implementation of the filter, we choose the subdivision that leads to the first grid point to be at 0. In this subdivision, the first area starts at $2\pi - \frac{\pi}{n}$. Thus, assuming $n > 3$, we obtain the subdivision $\mathcal{A} = \{[2\pi - \frac{\pi}{n}, \frac{\pi}{n}), [\frac{\pi}{n}, 3 \cdot \frac{\pi}{n}), [3 \cdot \frac{\pi}{n}, 5 \cdot \frac{\pi}{n}), \ldots, [2\pi - 3 \cdot \frac{\pi}{n}, 2\pi - \frac{\pi}{n})\}$ and the grid points $\mathcal{B} = \{0, \frac{2\pi}{n}, \frac{4\pi}{n}, \ldots, (n-2) \cdot \frac{\pi}{n}\}$. We use this subdivision in our filter because all grid points can be represented as $2\pi\frac{k}{n}$ with $k \in \{0, \ldots, n-1\}$.

We proceed by providing the full details on the assumptions of our filters that we briefly summarized in the introduction. If all assumptions hold, all densities and likelihoods involved can be represented accurately and the update step will be precise (as we later see, the prediction step involves additional assumptions and approximations). To each grid point β_i, we assign a grid value, which we denote by γ_i. There are different ways to interpret the grid values [23]. However, we will not go into detail on this in our current paper and merely focus on the interpretation that γ_i describes the function value of $f^\omega(\cdot)$ in the area A_i.

This leads us to the assumptions (A1)–(A3) that underlie the density representation in this paper.

(A1) The function value in each area is identical to the corresponding grid value, i.e., $\forall \alpha \in A_i : f^\omega(\alpha) = \gamma_i$.
(A2) The conditional density $f^c(\underline{x}^\tau | \alpha)$ is the same for all $\alpha \in A_i$ (we denote it by $f_i^c(\underline{x}^\tau)$).
(A3) The conditional density $f^c(\underline{x}^\tau | \alpha)$ is Gaussian for every considered α.

By combining assumptions (A2) and (A3), we get the assumption that every $f_i^c(\underline{x}^\tau)$ is Gaussian. We shall call the mean of that Gaussian $\underline{\mu}_i$ and the covariance $\mathbf{C}_i$, and write the pdf of the normal distribution $\mathcal{N}(\underline{\mu}_i, \mathbf{C}_i)$ at $\underline{x}^\tau$ as $f_{\mathrm{N}}(\underline{x}^\tau; \underline{\mu}_i, \mathbf{C}_i)$. The combination of all three assumptions leads to a joint density consisting of scaled Gaussians that are extruded along the periodic dimension, as can be seen in Figure 2. For densities encountered in many scenarios, the assumptions (A1)–(A3) do not hold. By approximating densities and likelihoods based on these assumptions, we introduce errors. However, the errors tend to get smaller with an increasing number of regions n.

Based on these assumptions, we can now derive a formula for the parametric density. We start with the first two assumptions and then also introduce the third one, leading to

$$f\left(\underline{x}^\tau, x^\omega; \left\{(\gamma_i, \underline{\mu}_i, \mathbf{C}_i)\right\}_{i=1}^n\right) \overset{\text{(A1)\&(A2)}}{=} \sum_{i=1}^n \mathbb{1}\{x^\omega \in A_i\} \gamma_i f_i^c(\underline{x}^\tau) \tag{2}$$

$$\overset{\text{(A3)}}{=} \sum_{i=1}^n \mathbb{1}\{x^\omega \in A_i\} \gamma_i f_{\mathrm{N}}\left(\underline{x}^\tau; \underline{\mu}_i, \mathbf{C}_i\right), \tag{3}$$

with $\{(\gamma_i, \underline{\mu}_i, \mathbf{C}_i)\}_{i=1}^n$ denoting the set containing all tuples $(\gamma_i, \underline{\mu}_i, \mathbf{C}_i)$ for $i \in \{1, \ldots, n\}$ and $\mathbb{1}$ being an indicator function that is 1 when the expression in braces is true and 0 otherwise. Throughout this paper, we will use that the marginalized density for the periodic part does not depend on the parameters of the Gaussians. This can be proven via

$$
\begin{aligned}
f^{\omega}\left(x^{\omega};\left\{(\gamma_i,\underline{\mu}_i,\mathbf{C}_i)\right\}_{i=1}^{n}\right) &= \int_{\mathbb{R}^2}\sum_{i=1}^{n}\mathbb{1}\{x^{\omega}\in A_i\}\gamma_i f_{\mathrm{N}}\left(\underline{x}^{\tau};\underline{\mu}_i,\mathbf{C}_i\right)d\underline{x}^{\tau}\\
&= \sum_{i=1}^{n}\mathbb{1}\{x^{\omega}\in A_i\}\gamma_i\int_{\mathbb{R}^2} f_{\mathrm{N}}\left(\underline{x}^{\tau};\underline{\mu}_i,\mathbf{C}_i\right)d\underline{x}^{\tau} \qquad (4)\\
&= \sum_{i=1}^{n}\mathbb{1}\{x^{\omega}\in A_i\}\gamma_i\,.
\end{aligned}
$$

Thus, we can also write the marginal as $f^{\omega}\left(x^{\omega};\underline{\gamma}\right)$, with $\underline{\gamma}$ being the vector containing all grid values.

In the remainder of this section, we first address how we can obtain the parameters for a given density. The second subsection addresses how we can calculate the parameter of a normalized density from the parameters describing an unnormalized one. In the third subsection, we describe other ways to provide continuous densities based on the same parameter set. This only serves as a means to obtain smoother densities than the one in Figure 2 and is not used in the prediction and update steps.

2.1. Deriving Parameters for a Given Density

We first discuss the case when the position and orientation are independent in the next paragraph and then proceed with the general case. The position and orientation may be independent for the initial prior density or certain likelihood functions. Note that the orientation and position usually do not stay independent throughout the prediction and filter steps. If they did, separate filters could be used and the S3F would not be required.

If the position and orientation are independent, we have $f^{\mathrm{c}}(\underline{x}^{\tau}|x^{\omega})f^{\omega}(x^{\omega}) = f^{\tau}(\underline{x}^{\tau})f^{\omega}(x^{\omega})$. For the periodic part $f^{\omega}(x^{\omega})$, the function values at the grid points $\mathcal{B}$ are stored in the vector of grid values $\check{\underline{\gamma}}$, as is done by the grid filter for the circle [19]. If $f^{\tau}(\underline{x}^{\tau})$ is a Gaussian density, all $\underline{\mu}_i$ and $\mathbf{C}_i$ are set to its parameters (the parameters are the same for each area due to the independence of $f^{\tau}(\underline{x}^{\tau})$ and $f^{\omega}(x^{\omega})$). If $f^{\tau}(\underline{x}^{\tau})$ is not a Gaussian, we can approximate it with a Gaussian, e.g., via moment matching.

We now consider the case in which the two parts are dependent. We do not require the full density $f^{\omega}(x^{\omega})$ but only the value of the density at the grid points. The values can be determined via

$$
\check{\gamma}_i = f^{\omega}(\beta_i) = \int_{\mathbb{R}^2} f(\underline{x}^{\tau},\beta_i)d\underline{x}^{\tau}\,, \qquad (5)
$$

which involves a 2-D integral for each of the n areas. Based on the grid values, we obtain the conditional density

$$
f_i^{\mathrm{c}}(\underline{x}^{\tau}) = f^{\mathrm{c}}(\underline{x}^{\tau}|\beta_i) = \frac{f(\underline{x}^{\tau},\beta_i)}{f^{\omega}(\beta_i)} = \frac{1}{\check{\gamma}_i}f(\underline{x}^{\tau},\beta_i) \qquad (6)
$$

for every region. If the parameters of the Gaussian are not directly available, we can use

$$
\underline{\mu}_i = \int_{\mathbb{R}^2}\underline{x}^{\tau}f_i^{\mathrm{c}}(\underline{x}^{\tau})d\underline{x}^{\tau} \qquad (7)
$$

to obtain the mean vector of the Gaussian for the area i and

$$
c_{i,j,k} = \int_{\mathbb{R}^2}(x_i^{\tau}-\mu_{i,j})(x_j^{\tau}-\mu_{j,k})f_i^{\mathrm{c}}(\underline{x}^{\tau})d\underline{x}^{\tau} \qquad (8)
$$

for the entry in column j and row k of the covariance matrix for area i. In total, we need n 2-D integrals for the grid values, $2n$ 2-D integrals for all components of the means for all areas, and $3n$ 2-D integrals for all components of the covariance matrices. Due to the expensive integrals, determining the parameters should be avoided during prediction and update steps. However, the integral formulae can be useful for, e.g., transforming a prior density offline.

The grid values are guaranteed to be nonnegative since they are function values of $f^{\omega}(x^{\omega})$ or integrals of function values of $f(\underline{x}^{\tau}, x^{\omega})$. However, it cannot be guaranteed that any of the grid values is positive. If none is positive, using a higher resolution or a different filter should be considered. Furthermore, the density may not be normalized. If $f^{\omega}(x^{\omega}; \underline{\breve{\gamma}})$ is normalized, then the joint density based on the parameters is also normalized since the Gaussian for the translation is inherently normalized. However, it may happen that $f^{\omega}(x^{\omega}; \underline{\breve{\gamma}})$ is unnormalized. This is the reason why we wrote the vector as $\underline{\breve{\gamma}}$, and we will continue using this decorator to denote that the density described by the vector may be unnormalized. The final step of our approximation procedure is to determine the parameters of the normalized density. This is done as described in the next subsection.

2.2. Normalization

For normalizing a density, it is crucial how a parametric density as described by (3) can be scaled. It is trivial to see that if one scales every grid value with a factor λ, all function values are scaled by λ. The factor by which we need to scale the function is the reciprocal of the integral since scaling it with this value will lead to a function that integrates to one. We first derive the formula for the integral of the parametric density. For this, we use (4) to obtain

$$\frac{1}{\lambda} = \int_{\mathbb{R}^2} \int_{\mathbb{S}} f\left(\underline{x}^{\tau}, x^{\omega}; \left\{(\breve{\gamma}_i, \underline{\mu}_i, \mathbf{C}_i)\right\}_{i=1}^{n}\right) dx^{\omega} d\underline{x}^{\tau} = \int_{\mathbb{S}} f^{\omega}\left(x^{\omega}; \underline{\breve{\gamma}}\right) dx^{\omega} \,. \tag{9}$$

With this formula, we can confirm that the joint density is normalized when $f^{\omega}\left(x^{\omega}; \underline{\breve{\gamma}}\right)$ is normalized.

We can now proceed using the assumption (A1) to get

$$\frac{1}{\lambda} = \int_{\mathbb{S}} \sum_{i=1}^{n} \mathbb{1}\{x^{\omega} \in A_i\} \breve{\gamma}_i dx^{\omega} = \sum_{i=1}^{n} \breve{\gamma}_i |A_i| = \frac{2\pi}{n} \sum_{i=1}^{n} \breve{\gamma}_i = \frac{2\pi}{n} \left\| \underline{\breve{\gamma}} \right\|_1 \tag{10}$$

in which $|A_i|$ is the size of the respective subset of the periodic domain A_i and $\|\cdot\|_1$ denotes the 1-norm of the vector. Hence, the vector describing a normalized density can be obtained via

$$\underline{\gamma} = \underline{\breve{\gamma}} / \left(\frac{2\pi}{n} \left\| \underline{\breve{\gamma}} \right\|_1 \right) . \tag{11}$$

2.3. Providing Continuous Densities

One way to provide a continuous density is to use (3), which integrates all assumptions on which we base the prediction and update steps. Since modulo arithmetics can be employed to find the relevant area in $O(1)$ and evaluating the corresponding Gaussian is also in $O(1)$, evaluating the density is in $O(1)$. We can also apply (3) for Cartesian products of $\mathbb{R}^d$ with other periodic (or bounded) domains.

In Figure 2, we illustrate how an approximation according to (3) looks like. While the result clearly resembles the original density, it is not an accurate approximation. In our first modification to obtain a density that is closer to the original density, we drop the assumption (A1). In the paper on the grid filter [19] that our current paper is based on, two ways to interpolate a periodic function based on function values on a grid are given. The first one does not ensure the nonnegativity of the density, while the other one does. We use the interpolation that ensures the nonnegativity and apply it to the periodic part $f^{\omega}(x^{\omega})$. As stated previously, the joint density is normalized if the density for the periodic part is normalized. Thus, the normalization of the density when changing the interpolation scheme for the periodic part can be directly concluded from the normalization of the density for the periodic part, which is proven in the appendix of [19]. In Figure 3, we show the result when the conditional densities are unchanged. The density is now much smoother but still shows significant changes at the borders of the different areas and is not yet an accurate approximation of the original density.

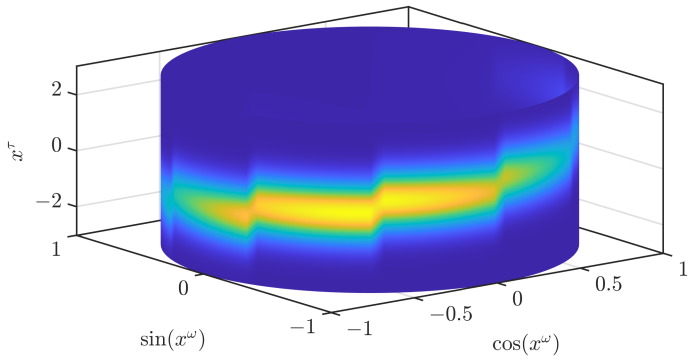

Figure 3. Continuous density based on a trigonometric polynomial for the periodic part.

To obtain smooth transitions between the areas, we drop the assumption (A2) and consider potentially (different) conditionals within each area. Since we only have parameters of n Gaussians, we explore ways to generate other conditionals based on these parameters. For the conditional at a grid point β_i, the parameters $\underline{\mu}_i$ and $\mathbf{C}_i$ should always be used. For any other point α, we determine the points β_i and β_{i+1} (for $i = n$, $i+1$ shall be 1) that are closest to α in respect to the distance d_0. Then, we calculate a value that describes how much of the distance between two consecutive grid points (which is $2\pi/n$) lies between β_i and α using

$$\eta(\alpha) = \frac{d(\alpha, \beta_i)}{2\pi n} \,. \tag{12}$$

The corresponding factor for β_{i+1} is $1 - \eta(\alpha)$.

As the first of two alternative ways to obtain other conditionals, we propose to also drop (A3) and use the Gaussian mixture

$$f^{\mathrm{c}}(\underline{x}^\tau|\alpha) = \eta f_i^{\mathrm{c}}(\underline{x}^\tau) + (1-\eta) f_{i+1}^{\mathrm{c}}(\underline{x}^\tau) \tag{13}$$

instead of the Gaussian at the closest grid point. This leads to a much smoother result, as can be seen in Figure 4. However, $f^{\mathrm{c}}(\underline{x}^\tau|\alpha)$ may be multimodal for $\alpha \notin \mathcal{B}$.

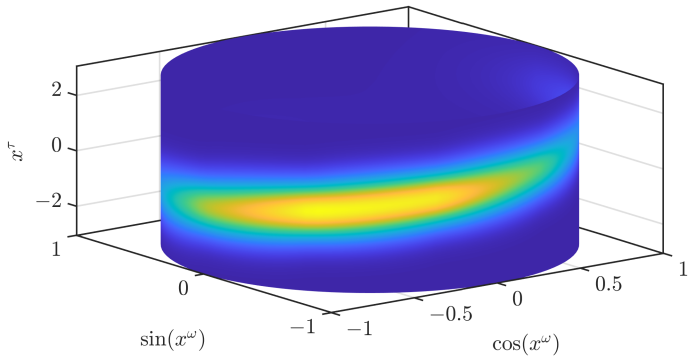

Figure 4. Continuous density obtained by the use of a trigonometric polynomial for the periodic part and distance-based mixtures of Gaussians for the linear part.

Our second option is to preserve (A3) and use a convex combination of the Gaussians' parameters as the parameters of a new Gaussian according to

$$\underline{\mu}(\alpha) = \eta(\alpha)\underline{\mu}_i + (1-\eta(\alpha))\underline{\mu}_{i+1} \,, \qquad \mathbf{C}(\alpha) = \eta(\alpha)\mathbf{C}_i + (1-\eta(\alpha))\mathbf{C}_{i+1} \,. \tag{14}$$

This guarantees that the conditional density of $\mathcal{N}(\underline{\mu}(\alpha), \mathbf{C}(\alpha))$ is always unimodal. In the formula for the covariance, we use that the convex combination of symmetric positive definite matrices is always symmetric positive definite. While this approach leads to a good approximation quality, we did not observe a clear superiority over using (13). For our considered example, the plot is visually indistinguishable from Figure 4, which is why we

have omitted it. This approach may be more expensive in practical implementations since a different covariance matrix is used for every point at which the pdf is to be evaluated. The other two approaches involve evaluating at most n differently parameterized Gaussian densities, regardless of the number of points at which the pdf should be evaluated.

3. Filter Derivation

In this section, we derive the update and prediction steps of the S3F. A formula that we will require in this section is the fundamental Gaussian identity (App. D, [26])

$$\begin{aligned} f_{\mathrm{N}}(\mathbf{M}\underline{x};\underline{\mu}^r,\mathbf{C}^r)f_{\mathrm{N}}(\underline{x};\underline{\mu}^s,\mathbf{C}^s) &= f_{\mathrm{N}}(\underline{\mu}^r;\mathbf{M}\underline{\mu}^s,\mathbf{C}^r+\mathbf{M}\mathbf{C}^s\mathbf{M}^\top)f_{\mathrm{N}}(\underline{x};\underline{\mu}^q,\mathbf{C}^q)\,, \\ \mathbf{C}^q = \left(\mathbf{M}^\top(\mathbf{C}^r)^{-1}\mathbf{M}+(\mathbf{C}^s)^{-1}\right)^{-1}\,,&\quad \underline{\mu}^q = \mathbf{C}^q\left(\mathbf{M}^\top(\mathbf{C}^r)^{-1}\underline{\mu}^r+(\mathbf{C}^s)^{-1}\underline{\mu}^s\right)\,, \end{aligned} \tag{15}$$

which can be applied for symmetric positive definite matrices $\mathbf{C}^r$ and $\mathbf{C}^s$, the matrix $\mathbf{M}$, and vectors $\underline{\mu}^r$, $\underline{\mu}^s$, and $\underline{x}$ with compatible sizes.

3.1. Update Step

The update step is performed to improve our parameters based on measurements. For simplicity, we say a single measurement is obtained at time step t and denote it by $\hat{\underline{z}}_t$. In our filter, we work with measurement models in the form of a likelihood function $\mathcal{L}_{\hat{\underline{z}}_t}(\underline{x}_t^\tau, x_t^\omega)$. The likelihood function (which is not necessarily a density) describes the probability that the measurement $\hat{\underline{z}}_t$ is obtained when the state is $[\underline{x}_t^\tau, x_t^\omega]$ (we use ; in brackets to denote that a vector is stacked vertically). The update step is based on Bayes' rule, leading to

$$\begin{aligned} f_t^{\mathrm{e}}(\underline{x}_t^\tau, x_t^\omega|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t) &= \frac{\mathcal{L}_{\hat{\underline{z}}_t}(\underline{x}_t^\tau, x_t^\omega)f_t^{\mathrm{p}}(\underline{x}_t^\tau, x_t^\omega|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_{t-1})}{\int_{\mathbb{S}}\int_{\mathbb{R}^2}\mathcal{L}_{\hat{\underline{z}}_t}(\underline{x}_t^\tau, x_t^\omega)f_t^{\mathrm{p}}(\underline{x}_t^\tau, x_t^\omega|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_{t-1})\,d\underline{x}_t^\tau dx_t^\omega} \\ &\propto \underbrace{\mathcal{L}_{\hat{\underline{z}}_t}(\underline{x}_t^\tau, x_t^\omega)f_t^{\mathrm{p}}(\underline{x}_t^\tau, x_t^\omega|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_{t-1})}_{\check{f}_t^{\mathrm{e}}(\underline{x}_t^\tau, x_t^\omega|\hat{\underline{z}}_1,\ldots,\hat{\underline{z}}_t)}\,. \end{aligned} \tag{16}$$

with f_t^{e} denoting the posterior and f_t^{p} the prior density. We now focus on obtaining the unnormalized prior $\check{f}_t^{\mathrm{e}}$, which we can easily normalize, as explained in Section 2.2.

To simplify the notation, we omit all time indices (in the update step, it is always t) except in the index of the state and measurement. We also omit conditioning on measurements for additional brevity (refer to (16) to see which density is conditioned on which measurements). In the previous prediction step (or the initialization), we obtained the prior density

$$f^{\mathrm{p}}\left(\underline{x}_t^\tau, x_t^\omega, \left\{(\gamma_j,\underline{\mu}_j^{\mathrm{p}},\mathbf{C}_j^{\mathrm{p}})\right\}_{j=1}^n\right) = \sum_{j=1}^n \mathbb{1}\{x_t^\omega\in A_j\}\gamma_j^{\mathrm{p}} f_{\mathrm{N}}\left(\underline{x}_t^\tau;\underline{\mu}_j^{\mathrm{p}},\mathbf{C}_j^{\mathrm{p}}\right)\,. \tag{17}$$

For the likelihood function, we also introduce the assumptions (A1) that $\mathcal{L}^\omega_{\hat{\underline{z}}_t}(\alpha)=\gamma_i$ for all $\alpha\in A_i$, (A2) that $\mathcal{L}^{\mathrm{c}}_{\hat{\underline{z}}_t}(\underline{x}_t^\tau|x_t^\omega=\alpha)=\mathcal{L}^{\mathrm{c}}_{\hat{\underline{z}}_t}(\underline{x}_t^\tau|x_t^\omega=\beta_i)$ for all $\alpha\in A_i$, and (A3) that all conditionals are Gaussians, leading to

$$\begin{aligned} \mathcal{L}_{\hat{\underline{z}}_t}\left(\underline{x}_t^\tau, x_t^\omega; \left\{(\gamma_i;\underline{\mu}_i^{\mathrm{L}},\mathbf{C}_i^{\mathrm{L}})\right\}_{i=1}^n\right) &\overset{\text{(A1)\&(A2)}}{=} \sum_{i=1}^n \mathbb{1}\{x_t^\omega\in A_i\}\gamma_i\mathcal{L}^{\mathrm{c}}_{\hat{\underline{z}}_t}(\underline{x}_t^\tau|x_t^\omega=\beta_i) \\ &\overset{\text{(A3)}}{=} \sum_{i=1}^n \mathbb{1}\{x_t^\omega\in A_i\}\gamma_i f_{\mathrm{N}}\left(\underline{x}_t^\tau;\underline{\mu}_i^{\mathrm{L}},\mathbf{C}_i^{\mathrm{L}}\right)\,. \end{aligned} \tag{18}$$

Since the assumptions may not be fulfilled in general, $\mathcal{L}_{\hat{\underline{z}}_t}\left(\underline{x}_t^\tau, x_t^\omega; \left\{(\gamma_i;\underline{\mu}_i^{\mathrm{L}},\mathbf{C}_i^{\mathrm{L}})\right\}_{i=1}^n\right)$ will generally not be equal to the original likelihood $\mathcal{L}_{\hat{\underline{z}}_t}(\underline{x}_t^\tau, x_t^\omega)$. We assume we have the grid values $\gamma_i=\mathcal{L}^\omega_{\hat{\underline{z}}_t}(\beta_i)$ or have the marginalized density for the periodic part $\mathcal{L}^\omega_{\hat{\underline{z}}_t}(x_t^\omega)$ and

can evaluate it at any point. If integrals are required to determine (or approximate) any of the parameters, this has a significant negative impact on the run times. However, in many settings (see, e.g., our evaluation scenario in Section 4.1) integrals are not required to obtain the likelihood function.

The likelihood function is special in multiple ways. First, while it is nonnegative everywhere, it may not be and does not need to be normalized. The multiplication of the likelihood and the prior density is generally unnormalized, and thus, a normalization step is anyway required at the end of the update step. Second, the likelihood may only depend on a certain part of the state. If the likelihood does not depend on the linear part, one can just update the grid values as in the regular grid filter [19] without adapting the parameters of the Gaussians. If the likelihood does not depend on the orientation part, we set all γ_i to 1. It is important to still perform the update step as described in this subsection and not merely perform Kalman filter updates for updating the Gaussians' parameters in each region. This will become evident in our derivation.

We first introduce the assumptions (A1) and (A2) to obtain

$$\begin{aligned} \breve{f}^{\mathrm{e}}(\underline{x}_t^{\tau}, x_t^{\omega}) &\overset{\text{(A1)\&(A2)}}{=} \left(\sum_{i=1}^{n} \mathbb{1}\{x_t^{\omega} \in A_i\} \gamma_i^{\mathrm{L}} \mathcal{L}_{\hat{\underline{x}}_t}^{\mathrm{c}}(\underline{x}_t^{\tau} | x_t^{\omega} = \beta_i)\right) \left(\sum_{j=1}^{n} \mathbb{1}\{x_t^{\omega} \in A_j\} \gamma_j^{\mathrm{p}} f_j^{\mathrm{p},c}(\underline{x}_t^{\tau})\right) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{1}\{x_t^{\omega} \in A_i\} \mathbb{1}\{x_t^{\omega} \in A_j\} \gamma_i^{\mathrm{L}} \gamma_j^{\mathrm{p}} \mathcal{L}_{\hat{\underline{x}}_t}^{\mathrm{c}}(\underline{x}_t^{\tau} | x_t^{\omega} = \beta_i) f_j^{\mathrm{p},c}(\underline{x}_t^{\tau}) \\ &= \sum_{i=1}^{n} \mathbb{1}\{x_t^{\omega} \in A_i\} \gamma_i^{\mathrm{L}} \gamma_i^{\mathrm{p}} \mathcal{L}_{\hat{\underline{x}}_t}^{\mathrm{c}}(\underline{x}^{\tau} | x^{\omega} = \beta_i) f_i^{\mathrm{p},c}(\underline{x}^{\tau}) \,. \end{aligned} \tag{19}$$

Then, we use (A3) and (15), leading to

$$\begin{aligned} \breve{f}^{\mathrm{e}}(\underline{x}_t^{\tau}, x_t^{\omega}) &\overset{\text{(A3)}}{=} \sum_{i=1}^{n} \mathbb{1}\{x_t^{\omega} \in A_i\} \gamma_i^{\mathrm{L}} \gamma_i^{\mathrm{p}} f_{\mathrm{N}}\left(\underline{x}_t^{\tau}; \underline{\mu}_i^{\mathrm{L}}, \mathbf{C}_i^{\mathrm{L}}\right) f_{\mathrm{N}}\left(\underline{x}_t^{\tau}; \underline{\mu}_i^{\mathrm{p}}, \mathbf{C}_i^{\mathrm{p}}\right) \\ &\overset{(15)}{=} \sum_{i=1}^{n} \mathbb{1}\{x_t^{\omega} \in A_i\} \underbrace{\gamma_i^{\mathrm{L}} \gamma_i^{\mathrm{p}} f_{\mathrm{N}}\left(\underline{\mu}_i^{\mathrm{p}}; \underline{\mu}_i^{\mathrm{L}}, \mathbf{C}_i^{\mathrm{L}} + \mathbf{C}_i^{\mathrm{p}}\right)}_{\breve{\gamma}_i} f_{\mathrm{N}}\left(\underline{x}_t^{\tau}; \underline{\mu}_i^{\mathrm{e}}, \mathbf{C}_i^{\mathrm{e}}\right) \end{aligned} \tag{20}$$

with

$$\mathbf{C}_i^{\mathrm{e}} = \left((\mathbf{C}_i^{\mathrm{L}})^{-1} + (\mathbf{C}_i^{\mathrm{p}})^{-1}\right)^{-1}, \tag{21}$$

$$\underline{\mu}_i^{\mathrm{e}} = \mathbf{C}_i^{\mathrm{e}}((\mathbf{C}_i^{\mathrm{L}})^{-1} \underline{\mu}_i^{\mathrm{L}} + (\mathbf{C}_i^{\mathrm{p}})^{-1} \underline{\mu}_i^{\mathrm{p}}) \,. \tag{22}$$

As the parameters of the unnormalized posterior density, we thus obtain the parameters of the Gaussians according to (21) and (22) and the grid values using

$$\breve{\gamma}_i^{\mathrm{e}} = \gamma_i^{\mathrm{L}} \gamma_i^{\mathrm{p}} f_{\mathrm{N}}\left(\underline{\mu}_i^{\mathrm{p}}; \underline{\mu}_i^{\mathrm{L}}, \mathbf{C}_i^{\mathrm{L}} + \mathbf{C}_i^{\mathrm{p}}\right) . \tag{23}$$

The parameters of the normalized posterior density can be calculated as explained in Section 2.2 using $\underline{\gamma}^{\mathrm{e}} = \underline{\breve{\gamma}}^{\mathrm{e}} / \left(2\pi \left\| \underline{\breve{\gamma}}^{\mathrm{e}} \right\|_1\right)$. The Gaussians' parameters are unchanged by the normalization. From the formulae, we can see that the update step has a run time in $O(n)$.

No negative values can result from the formulae for the update step. However, it is possible that all new grid values are zero. In this case, using a higher grid resolution or employing a different filter should be attempted.

3.2. Prediction Step

The prediction step is based on a description of the system model in the form of a transition density $f_t^{\mathrm{T}}(\underline{x}_{t+1}^{\tau}, x_{t+1}^{\omega} | \underline{x}_t^{\tau}, x_t^{\omega})$, which describes the probability of transitioning

to state $[\underline{x}_{t+1}^{\tau}; x_{t+1}^{\omega}]$ when the current state is $[\underline{x}_{t}^{\tau}; x_{t}^{\omega}]$. The foundation for describing the predicted density for the next time step is the Chapman–Kolmogorov equation

$$f_{t+1}^{\mathrm{p}}(\underline{x}_{t+1}^{\tau}, x_{t+1}^{\omega} | \hat{\underline{z}}_1, \ldots, \hat{\underline{z}}_t) = \int_{\mathbb{R}^2} \int_{\mathbb{S}} \underbrace{f_t^{\mathrm{T}}(\underline{x}_{t+1}^{\tau}, x_{t+1}^{\omega} | \underline{x}_t^{\tau}, x_t^{\omega}) f_t^{\mathrm{e}}(\underline{x}_t^{\tau}, x_t^{\omega} | \hat{\underline{z}}_1, \ldots, \hat{\underline{z}}_t)}_{f_t^{\mathrm{j}}(\underline{x}_{t+1}^{\tau}, x_{t+1}^{\omega}, \underline{x}_t^{\tau}, x_t^{\omega})} d x_t^{\omega} d \underline{x}_t^{\tau} \tag{24}$$

for our state space. The formula on the right-hand side can be implemented as two consecutive operations. The first is to obtain the joint density f_t^{j} and the second is to marginalize x_t^{ω} and $\underline{x}_t^{\tau}$ out.

We now approximate the transition density using a suitable parametric density. As in the update step, we omit the time indices for the functions (refer to (24) for these) and the measurements on which the density is conditioned. We split the transition density into a marginalized density for the periodic part and a conditional density according to

$$f^{\mathrm{T}}(\underline{x}_{t+1}^{\tau}, x_{t+1}^{\omega} | \underline{x}_t^{\tau}, x_t^{\omega}) = f^{\mathrm{T,c,full}}(\underline{x}_{t+1}^{\tau} | x_{t+1}^{\omega}, \underline{x}_t^{\tau}, x_t^{\omega}) f^{\mathrm{T},\omega,\mathrm{full}}(x_{t+1}^{\omega} | \underline{x}_t^{\tau}, x_t^{\omega}) \,. \tag{25}$$

To arrive at a feasible prediction step, we now assume that the orientation at time step $t+1$ is conditionally independent of the position at time step t given the orientation at time step t. When this is the case, a transition density $f^{\mathrm{T},\omega}(x_{t+1}^{\omega} | x_t^{\omega})$ may be directly available. If only the full transition density is available, one can obtain $f^{\mathrm{T},\omega}$ by marginalizing the position at time step $t+1$ out of f^{T} and discarding the (irrelevant) dependency on the position at time step t. When the conditional independence does not hold, one may consider conditioning on a specific value for the position part, e.g., the current estimate of the filter or the mean vector in the respective area.

The key idea of the prediction step is to consider the transition density as a function (not a density) of $\mathbb{S} \times \mathbb{R}^2 \times \mathbb{S} \times \mathbb{R}^2$. By reordering the parameters, we can see it as a function of $\mathbb{S} \times \mathbb{S} \times \mathbb{R}^2 \times \mathbb{R}^2$ Then, we can consider the toroidal manifold $\mathbb{S} \times \mathbb{S}$ as the periodic domain that we subdivide. For compatibility with f^{e}, we subdivide $\mathbb{S} \times \mathbb{S}$ equally along both axes so that we obtain the areas $\{A_i \times A_j\}_{(i,j) \in \{1,\ldots,n\}^2}$. For each area $A_i \times A_j$, the corresponding grid point is $[\beta_i; \beta_j]$. Then, we can get a similar grid-based representation as previously discussed by introducing some assumptions that resemble (A1)–(A3). Combined with the assumption introduced in the previous paragraph, we arrive at the list of assumptions

(B1) $f^{\mathrm{T},\omega}(x_{t+1}^{\omega} | x_t^{\omega}) = f^{\mathrm{T},\omega,\mathrm{full}}(x_{t+1}^{\omega} | \underline{x}_t^{\tau}, x_t^{\omega})$ holds for all $\underline{x}_t^{\tau} \in \mathbb{R}^2$.

(B2) The function value of $f^{\mathrm{T},\omega}(x_{t+1}^{\omega} | x_t^{\omega})$ in each area in $\mathbb{S} \times \mathbb{S}$ is identical to the grid value for this area, i.e., $f^{\mathrm{T},\omega}(\alpha_1 | \alpha_2) = f^{\mathrm{T},\omega}(\beta_i | \beta_j) = \gamma_{i,j}^{\mathrm{T}}$ for all $\alpha_1 \in A_i$, $\alpha_2 \in A_j$.

(B3) The conditional density $f^{\mathrm{T,c,full}}(\underline{x}_{t+1}^{\tau} | \alpha_1, \underline{x}_t^{\tau}, \alpha_2)$ is the same function for all $\alpha_1 \in A_i$, $\alpha_2 \in A_j$ (we denote it by $f_{i,j}^{\mathrm{T,c}}(\underline{x}_{t+1}^{\tau} | \underline{x}_t^{\tau})$).

(B4) Each conditional density can be written as

$$f_{i,j}^{\mathrm{T,c}}(\underline{x}_{t+1}^{\tau} | \underline{x}_t^{\tau}) = f_{\mathrm{N}}\left(\underline{x}_{t+1}^{\tau}; \mathbf{F}_{t,i,j} \underline{x}_t^{\tau} + \underline{\hat{u}}_{t,i,j}, \mathbf{C}_{t,i,j}^{\underline{w}}\right) \tag{26}$$

and we have all relevant system matrices $\mathbf{F}_{t,i,j}$, inputs $\underline{\hat{u}}_{t,i,j}$, and covariance matrices $\mathbf{C}_{t,i,j}^{\underline{w}}$.

With (B3), we assume that for a fixed orientation, the transitional part follows a linear system model. Writing it using random vectors (written in bold font), the model can be described by

$$\underline{\boldsymbol{x}}_{t+1,i,j}^{\tau} = \mathbf{F}_{t,i,j} \underline{\boldsymbol{x}}_{t,i,j}^{\tau} + \underline{\hat{u}}_{t,i,j}^{\tau} + \underline{\boldsymbol{w}}_{t,i,j}^{\tau} \tag{27}$$

with fixed and known system input $\underline{\hat{u}}_{t,i,j}^{\tau}$ and system noise $\underline{\boldsymbol{w}}_{t,i,j} \sim \mathcal{N}(\underline{0}, \mathbf{C}_{t,i,j}^{\underline{w}})$. If $\underline{\boldsymbol{w}}_{t,i,j}$ is not a Gaussian, we propose to approximate it with one via moment matching. The input $\underline{\hat{u}}_{t,i,j}^{\tau}$ can be different for each area and can thus depend on the orientation. If it is a function in the form of $g(x_t^{\omega})$, one can use β_i as x_t^{ω}, i.e., $\underline{\hat{u}}_{t,i,j}^{\tau} = g(\beta_j)$. Similarly, the input can also

be a function of x_t^ω and x_{t+1}^ω. While we never explicitly consider inputs for the orientation part, they can be considered by the nonlinear system model described by $f^{\mathrm{T},\omega}$.

In our description of the transition density, we shall from now on omit the time indices of the input vector and the system and covariance matrices, which is always t. Based on its parameters, the transition density is given by

$$\begin{aligned} & f^{\mathrm{T}}\left(\underline{x}_{t+1}^\tau, x_{t+1}^\omega | \underline{x}_t^\tau, x_t^\omega; \left\{\gamma_{i,j}^{\mathrm{T}}, \mathbf{F}_{i,j}, \underline{\hat{u}}_{i,j}^\tau, \mathbf{C}_{i,j}^{\underline{w}}\right\}_{(i,j)\in\{1,\dots,n\}^2}\right) \\ \overset{\text{(B1)–(B2)}}{=} & \sum_{i=1}^n \sum_{j=1}^n \mathbb{1}\{x_{t+1}^\omega \in A_i\}\mathbb{1}\{x_t^\omega \in A_j\}\gamma_{i,j}^{\mathrm{T}} f^{\mathrm{T,c,full}}(\underline{x}_{t+1}^\tau | x_{t+1}^\omega, \underline{x}^\tau, x_t^\omega) \\ \overset{\text{(B3)}}{=} & \sum_{i=1}^n \sum_{j=1}^n \mathbb{1}\{x_{t+1}^\omega \in A_i\}\mathbb{1}\{x_t^\omega \in A_j\}\gamma_{i,j}^{\mathrm{T}} f_{i,j}^{\mathrm{T,c}}(\underline{x}_{t+1}^\tau | \underline{x}_t^\tau)\,. \end{aligned} \tag{28}$$

Due to potential approximations involved, the density may be unnormalized. By default, we renormalize after the prediction step (if necessary) and do not perform any normalization on the parametric transition density.

We start our derivation of the predicted density by writing out and reformulating the joint density as

$$\begin{aligned} & f^{\mathrm{j}}(\underline{x}_{t+1}^\tau, x_{t+1}^\omega, \underline{x}_t^\tau, x_t^\omega) \\ \overset{\text{(B2)\&(B3)}}{=} & \left(\sum_{i=1}^n \sum_{j=1}^n \mathbb{1}\{x_{t+1}^\omega \in A_i\}\mathbb{1}\{x_t^\omega \in A_j\}\gamma_{i,j}^{\mathrm{T}} f_{i,j}^{\mathrm{T,c}}(\underline{x}_{t+1}^\tau | \underline{x}_t^\tau)\right)\left(\sum_{k=1}^n \mathbb{1}\{x_t^\omega \in A_k\}\gamma_k^{\mathrm{e}} f_k^{\mathrm{e}}(\underline{x}_t^\tau)\right) \\ = & \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \mathbb{1}\{x_{t+1}^\omega \in A_i\}\mathbb{1}\{x_t^\omega \in A_j\}\mathbb{1}\{x_t^\omega \in A_k\}\gamma_{i,j}^{\mathrm{T}}\gamma_k^{\mathrm{e}} f_{i,j}^{\mathrm{T,c}}(\underline{x}_{t+1}^\tau | \underline{x}_t^\tau) f_k^{\mathrm{e}}(\underline{x}_t^\tau) \\ = & \sum_{i=1}^n \sum_{j=1}^n \mathbb{1}\{x_{t+1}^\omega \in A_i\}\mathbb{1}\{x_t^\omega \in A_j\}\gamma_{i,j}^{\mathrm{T}}\gamma_j^{\mathrm{e}} f_{i,j}^{\mathrm{T,c}}(\underline{x}_{t+1}^\tau | \underline{x}_t^\tau) f_j^{\mathrm{e}}(\underline{x}^\tau)\,. \end{aligned} \tag{29}$$

We define $\gamma_{i,j}^{\mathrm{j}} = \gamma_{i,j}^{\mathrm{T}}\gamma_j^{\mathrm{e}}$ and marginalize all components of the state at time step t out to obtain the predicted density

$$\begin{aligned} & f_{t+1}^{\mathrm{p}}(\underline{x}_{t+1}^\tau, x_{t+1}^\omega) \\ = & \int_{\mathbb{S}}\int_{\mathbb{R}^2} \sum_{i=1}^n \sum_{j=1}^n \mathbb{1}\{x_{t+1}^\omega \in A_i\}\mathbb{1}\{x_t^\omega \in A_j\}\gamma_{i,j}^{\mathrm{j}} f_{i,j}^{\mathrm{T,c}}(\underline{x}_{t+1}^\tau | \underline{x}_t^\tau) f_j^{\mathrm{e}}(\underline{x}_t^\tau) d\underline{x}_t^\tau dx_t^\omega \\ = & \sum_{i=1}^n \mathbb{1}\{x_{t+1}^\omega \in A_i\} \sum_{j=1}^n \int_{\mathbb{R}^2} \gamma_{i,j}^{\mathrm{j}} f_{i,j}^{\mathrm{T,c}}(\underline{x}_{t+1}^\tau | \underline{x}_t^\tau) f_j^{\mathrm{e}}(\underline{x}_t^\tau) d\underline{x}^\tau \int_{\mathbb{S}} \mathbb{1}\{x_t^\omega \in A_j\} dx_t^\omega \\ = & \sum_{i=1}^n \mathbb{1}\{x_{t+1}^\omega \in A_i\} |A_i| \sum_{j=1}^n \int_{\mathbb{R}^2} \gamma_{i,j}^{\mathrm{j}} f_{i,j}^{\mathrm{T,c}}(\underline{x}_{t+1}^\tau | \underline{x}_t^\tau) f_j^{\mathrm{e}}(\underline{x}_t^\tau) dx_t^\omega\,. \end{aligned} \tag{30}$$

We now introduce the assumption (B4) and use that the value of the pdf of a Gaussian at a point only depends on the distance between the point and the mode to rearrange the parameters of the first Gaussian. This allows us to employ (15). Thus, we get

$$
\begin{aligned}
& f_{t+1}^{\mathrm{p}}\left(\underline{x}_{t+1}^{\tau}, x_{t+1}^{\omega}\right) \\
& \overset{(B4)}{=} \sum_{i=1}^{n} \mathbb{1}\left\{x_{t+1}^{\omega} \in A_i\right\}|A_i| \sum_{j=1}^{n} \int_{\mathbb{R}^2} \gamma_{i,j}^{\mathrm{j}} f_{\mathrm{N}}\left(\underline{x}_{t+1}^{\tau}; \mathbf{F}_{i,j}\underline{x}_{t}^{\tau} + \underline{\hat{u}}_{i,j}^{\tau}, \mathbf{C}_{i,j}^{\underline{w}}\right) f_{\mathrm{N}}\left(\underline{x}_{t}^{\tau}; \underline{\mu}_{j}^{\mathrm{e}}, \mathbf{C}_{j}^{\mathrm{e}}\right) d\underline{x}_{t}^{\tau} \\
& = \sum_{i=1}^{n} \mathbb{1}\left\{x_{t+1}^{\omega} \in A_i\right\}|A_i| \sum_{j=1}^{n} \int_{\mathbb{R}^2} \gamma_{i,j}^{\mathrm{j}} f_{\mathrm{N}}\left(\mathbf{F}_{i,j}\underline{x}_{t}^{\tau}; \underline{x}_{t+1}^{\tau} - \underline{\hat{u}}_{i,j}^{\tau}, \mathbf{C}_{i,j}^{\underline{w}}\right) f_{\mathrm{N}}\left(\underline{x}_{t}^{\tau}; \underline{\mu}_{j}^{\mathrm{e}}, \mathbf{C}_{j}^{\mathrm{e}}\right) d\underline{x}_{t}^{\tau} \\
& \overset{(15)}{=} \sum_{i=1}^{n} \mathbb{1}\left\{x_{t+1}^{\omega} \in A_i\right\}|A_i| \sum_{j=1}^{n} \int_{\mathbb{R}^2} \gamma_{i,j}^{\mathrm{j}} f_{\mathrm{N}}\left(\underline{x}_{t+1}^{\tau} - \underline{\hat{u}}_{i,j}^{\tau}; \mathbf{F}_{i,j}\underline{\mu}_{j}^{\mathrm{e}}, \mathbf{C}_{i,j}^{\underline{w}} + \mathbf{F}_{i,j}\mathbf{C}_{j}^{\mathrm{e}}\mathbf{F}_{i,j}^{\top}\right) \\
& \qquad\qquad f_{\mathrm{N}}\left(\underline{x}_{t}^{\tau}; \underline{\mu}^{q}, \mathbf{C}^{q}\right) d\underline{x}_{t}^{\tau} \,,
\end{aligned} \tag{31}
$$

in which the precise values of $\underline{\mu}^{q}$ and $\mathbf{C}^{q}$ are irrelevant. Since $f_{\mathrm{N}}\left(\underline{x}_{t}^{\tau}; \underline{\mu}^{q}, \mathbf{C}^{q}\right)$ is the only term that depends on $\underline{x}_{t}^{\tau}$, we can push back the integral sign to that term. Then, it is evident that the integral is 1 because we integrate a Gaussian density over the whole domain.

By adapting the parameters of the remaining Gaussian density in a way that does not change the function value, we obtain

$$
f_{t+1}^{\mathrm{p}}\left(\underline{x}_{t+1}^{\tau}, x_{t+1}^{\omega}\right) = \sum_{i=1}^{n} \mathbb{1}\left\{x_{t+1}^{\omega} \in A_i\right\}|A_i| \sum_{j=1}^{n} \gamma_{i,j}^{\mathrm{j}} f_{\mathrm{N}}\left(\underline{x}_{t+1}^{\tau}; \mathbf{F}_{i,j}\underline{\mu}_{j}^{\mathrm{e}}, +\underline{\hat{u}}_{i,j}^{\tau}, \mathbf{C}_{i,j}^{\underline{w}} + \mathbf{F}_{i,j}\mathbf{C}_{j}^{\mathrm{e}}\mathbf{F}_{i,j}^{\top}\right). \tag{32}
$$

Finally, we pull the factor $\left\|\underline{\gamma}_{i,:}^{\mathrm{j}}\right\|_1 = \sum_{k=1}^{n} \gamma_{i,k}^{\mathrm{j}}$ out of the second sum to obtain

$$
f_{t+1}^{\mathrm{p}}\left(\underline{x}_{t+1}^{\tau}, x_{t+1}^{\omega}\right) = \sum_{i=1}^{n} \mathbb{1}\left\{x_{t+1}^{\omega} \in A_i\right\} \overbrace{|A_i|\left\|\underline{\gamma}_{i,:}^{\mathrm{j}}\right\|_1}^{\gamma_i^{\mathrm{p}}} \sum_{j=1}^{n} \frac{\gamma_{i,j}^{\mathrm{j}}}{\left\|\underline{\gamma}_{i,:}^{\mathrm{j}}\right\|_1} f_{\mathrm{N}}\left(\underline{x}_{t+1}^{\tau}; \mathbf{F}_{i,j}\underline{\mu}_{j}^{\mathrm{e}} + \underline{\hat{u}}_{i,j}^{\tau}, \mathbf{C}_{i,j}^{\underline{w}} + \mathbf{F}_{i,j}\mathbf{C}_{j}^{\mathrm{e}}\mathbf{F}_{i,j}^{\top}\right). \tag{33}
$$

This is already reminiscent of the desired form (3). The factors in front of the second sum can be directly combined into the grid values for the predicted density $\underline{\gamma}^{\mathrm{p}}$. However, instead of a single Gaussian, we now have a mixture of Gaussians on $\mathbb{R}^2$ with the weights $\gamma_{i,j}^{\mathrm{j}} / \left\|\underline{\gamma}_{i,:}^{\mathrm{j}}\right\|_1$.

To prevent an increase in the number of parameters and an increase in the complexity of the filter over time, we perform mixture reduction. In our current implementation, we always reduce the mixture to a single Gaussian using moment matching. However, it would also be possible to perform a different reduction and preserve more components with alternative approaches [27]. Our use of moment matching is motivated by its speed and ease of implementation. Moment matching yields

$$
\underline{\mu}_{i}^{\mathrm{p}} = \sum_{j=1}^{n} \frac{\gamma_{i,j}^{\mathrm{j}}}{\left\|\underline{\gamma}_{i,:}^{\mathrm{j}}\right\|_1} \left(\mathbf{F}_{i,j}\underline{\mu}_{j}^{\mathrm{e}} + \underline{\hat{u}}_{i,j}^{\tau}\right) \tag{34}
$$

for the predicted mean of the ith area. For the covariance, we need to include both the part stemming from the individual covariances and the additional part caused by the differences in the means. For the area i, we obtain the formula

$$
\mathbf{C}_{i}^{\mathrm{p}} = \left(\sum_{j=1}^{n} \frac{\gamma_{i,j}^{\mathrm{j}}}{\left\|\underline{\gamma}_{i,:}^{\mathrm{j}}\right\|_1} \left(\mathbf{C}_{i,j}^{\underline{w}} + \mathbf{F}_{i,j}\mathbf{C}_{j}^{\mathrm{e}}\mathbf{F}_{i,j}^{\top}\right)\right) + \mathrm{SampleCov}\left(\left\{\mathbf{F}_{i,j}\underline{\mu}_{j}^{\mathrm{e}} + \underline{\hat{u}}_{i,j}^{\tau} - \underline{\mu}_{i}^{\mathrm{p}}\right\}_{j=1}^{n}, \frac{\gamma_{i,j}^{\mathrm{j}}}{\left\|\underline{\gamma}_{i,:}^{\mathrm{j}}\right\|_1}\right), \tag{35}
$$

with SampleCov denoting the sample covariance of the weighted zero mean (note that we subtract $\underline{\mu}_{i}^{\mathrm{p}}$) sample set. For a set of vectors $\{\underline{v}_1, \ldots, \underline{v}_n\}$ and a vector of weights $[w_1, \ldots, w_n]$, we obtain the values of the sample covariance according to

$$
c_{j,k}^{\mathrm{SampleCov}} = \sum_{l=1}^{n} w_l v_{j,l} v_{k,l} \,,
$$

in which $v_{j,l}$ is the lth component of the vector $\underline{v}_j$. Determining this sum is in $O(n)$. It has to be determined for all 3 (unique) entries of the covariance matrix.

Since both the mean and the covariance can be determined in $O(n)$, the total effort of the mixture reduction is in $O(n)$. Since n mixture reductions are required, the total effort involved to determine all new means and covariances is in $O(n^2)$. Calculating each of the n entries of $\underline{\gamma}^{\mathrm{p}}$ also involves summing over n values, and thus, the calculation of the grid values is also in $O(n^2)$. With both operations being in $O(n^2)$, the total effort of the prediction step is in $O(n^2)$.

4. Evaluation

In our evaluation, we consider a tracking scenario on $\mathrm{SE}(2)$ and compare our novel S3F with a particle filter (PF), the progressive $\mathrm{SE}(2)$ Bingham filter (ProgSE2BF) [14], and the unscented Kalman filter for manifolds (UKF-M).

The PF, ProgSE2BF, and S3F are part of the latest version of libDirectional [28]. For the UKF-M, we use the implementation of [13]. The UKF-M involves a parameter that can be tuned, which is called α in [29]. This parameter influences how far away from the mean the sigma points of the UKF [30] are placed. [13] recommends values in the range $[0.001, 1]$. The implementation allows specifying different parameters for the state uncertainty, the uncertainty introduced in the prediction step, and the measurement uncertainty. We set all entries of the vector to the same value and considered the values 0.001, 0.01, 0.1, and 1. While the results were similar for the different parameters, using 1 led to the best results. Therefore, we used this parameter in our evaluation.

4.1. Scenario Description

In the considered scenario, we track a vehicle for 50 time steps. Its initial position is distributed according to $\underline{x}_1^\tau \sim \mathcal{N}(\underline{0}, \mathbf{I})$ (with $\underline{0}$ denoting a zero vector and $\mathbf{I}$ an identity matrix), and its orientation (describing its heading angle) x_1^ω is distributed according to a von Mises distribution (Section 3.5.4, [25]) with $\mu = 0$ and $\kappa = 1$ (i.e., $x_1^\omega \sim \mathcal{VM}(0, 1)$) in each run.

The vehicle moves approximately one unit per time step in the direction it is facing. Random noise is applied to both the position and the orientation. The system model is thus

$$\begin{bmatrix} x_{t+1}^\omega \\ \underline{x}_{t+1}^\tau \end{bmatrix} = \begin{bmatrix} x_t^\omega + \underline{w}_t^\omega \mod 2\pi \\ \underline{x}_t^\tau + \begin{bmatrix} \cos(x_t^\omega) \\ \sin(x_t^\omega) \end{bmatrix} + \underline{w}_t^\tau \end{bmatrix} \tag{36}$$

with $\underline{w}_t^\tau \sim \mathcal{N}(\underline{0}, \mathbf{I})$ and $w_t^\omega \sim \mathcal{VM}(0, 10)$ for all $t \in \mathbb{N}$. The uncertainty in the orientation influences the uncertainty in the position as x_t^ω influences $\underline{x}_{t+1}^\tau$. Thus, while $\underline{w}_t^\tau$ and w_t^ω are independent, the noise in the angle influences later positions as $\underline{w}_t^\omega$ influences x_{t+1}^ω, which in turn influences $\underline{x}_{t+2}^\tau$. In each run, we drew the initial state from the initial prior and then generated the new states by applying the system model and sampling from the involved distributions. Thus, the path was different for each run. We provide four sample trajectories in Figure 5.

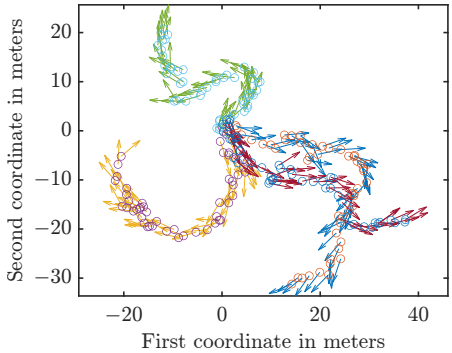

Figure 5. Sample trajectories. The circles indicate the positions, the arrows the heading angles.

We can only measure the position and not the heading angle of the vehicle. The measurement model is $\underline{z}_t = \underline{x}_t^\tau + \underline{v}_t$ with $\underline{v}_t \sim \mathcal{N}(\underline{0}, 0.5 \cdot \mathbf{I})$ for all $t \in \mathbb{N}$. This makes it challenging for filters since they have to derive information about the heading angle based on the dependency of the heading angle and position over time.

4.2. Models Used by the Filters

In this subsection, we describe how we applied the filters to the considered scenario. If possible, required conversions are done offline once and the results are used for all runs of the filter.

For the initialization of the PF with n particles, $\underline{x}_1^\tau$ and x_1^ω were sampled n times. These samples were then used as equally weighted samples for the state representation of the PF. For the ProgSE2BF, we draw 100,000 samples from $\underline{x}_1^\tau$ and x_1^ω and fit a density for dual quaternions states [31] to the samples.

Due to the independence of the distributions for the initial position and angle, we can directly store the function values on the grid in the vector $\underline{\gamma}_1^\mathrm{p}$ and initialize all Gaussians with $\underline{\mu}_{1,i}^\mathrm{p} = \underline{0}$ and $\mathbf{C}_{1,i}^\mathrm{p} = \mathbf{I}$ for $i \in \{1, \ldots, n\}$ for the S3F. For the UKF-M, we approximate the von Mises distribution with a wrapped normal (WN) distribution (Section 3.5.7, [25]) $\mathcal{WN}(0, \sigma_1^{\mathrm{p},\omega})$ via moment matching. The covariance matrix for the UKF-M is then a block diagonal matrix with the squared standard deviation $(\sigma_1^{\mathrm{p},\omega})^2$ of the WN and the covariance matrix $\mathbf{I}$ for the position on the diagonal.

Concerning the system model, we can sample the noise distributions for the PF and apply (36) (with samples instead of random variables) to the samples to obtain the samples of the predicted density. For the ProgSE2BF, we draw 100,000 samples from the system noise $[\underline{w}_t^\tau; \underline{w}_t^\omega]$ and fit a density for dual quaternion states to it. This has to be done only once offline because the resulting density can be used in all runs and time steps.

For the S3F, we evaluate $\mathcal{VM}(x_{t+1}^\omega; x_t^\omega, 1)$ on the two-dimensional grid to obtain the grid values $\gamma_{i,j}^\mathrm{T} = \mathcal{VM}(\beta_i; \beta_j, 1)$. This is done in a precomputation step offline. As the input $\underline{\hat{u}}_{t,i,j}^\tau$, we use $[\cos(\beta_j); \sin(\beta_j)]$, which describes the change in the position in one time step for the orientation β_j.

For the UKF-M, we use an existing model for tracking on $\mathrm{SE}(2)$ for scenarios involving gyroscopes. We provide a gyroscope reading of 1 m per time step (which is the true velocity when disregarding noise) to the model. For the system noise, we match w_t^ω with a WN distribution $\mathcal{WN}(0, \sigma^{w,\omega})$ and provide $\sigma^{w,\omega}$ to the filter. In the gyroscope-based model, the position noise needs to be given as longitudinal and transversal shifts. While our uncertainty in the position is given relative to the world coordinate system, our system covariance matrix for the position is $\mathbf{I}$, and this covariance matrix does not change under rotations of the coordinate system. Thus, we use $\mathbf{I}$ as the system covariance for the position for the UKF-M.

All filters except the UKF-M use likelihood functions. In the PF, the particle weights are multiplied with the function values of the likelihood at the particle positions. Afterward, a resampling step is performed. For the ProgSE2BF, we use the progressive update step that uses a likelihood function. The parameter for the progression (called τ in [14]) is set to 0.02. For the S3F, all grid values γ_i are set to 1 since the measurement does not provide any information on the orientation. The measurement $\underline{\hat{z}}_t$ is used as the mean $\underline{\mu}_i^\mathrm{L}$ for all Gaussians and $0.5 \cdot \mathbf{I}$ is used for all covariances $\mathbf{C}_i^\mathrm{L}$. A measurement function that extracts the state from the position is provided to the UKF-M. The employed covariance matrix is the covariance matrix of the measurement noise, i.e., $0.5 \cdot \mathbf{I}$.

4.3. Evaluation Metrics

For all filters, we measured the run times and divided them by the number of time steps to provide the run time per time step. The run times were measured on an Intel Xeon Gold 6230 running at 2.1 GHz with 3 GB of RAM allocated to the job. Matlab 2021a ran with a single thread on Red Hat Enterprise Linux 8.2.

To assess the estimation quality of the filters, we consider the accuracy of the position and orientation estimates. We consider them separately since one would have to specify a weighting factor when combining them, and any chosen value may appear arbitrary. Multiple runs were performed, and the errors and run times of all individual runs were condensed into a single value by determining their average.

For the PF, we discarded the part of the vector describing the orientation to marginalize it out and then determined the mean of the particles to obtain the position estimate. We did this likewise for the orientation component using the mean direction (Section 2.2.1, [25]) instead of the mean. For the S3F, we determined the mean of the linear part by marginalizing out the orientation part. This can be done analytically by determining the vector $\underline{\rho} = \underline{\gamma}_{50}^{\mathrm{e}} / \left\| \underline{\gamma}_{50}^{\mathrm{e}} \right\|_1$, weighting the Gaussians according to $\underline{\rho}$, and finally extracting the mean of the Gaussian mixture (see (34)). The circular mean of the weighted grid points is used as the estimate for the orientation.

The average Euclidean distance between the true and the estimated position is used as the position error. As error criterion for the periodic part, we use the topology-aware distance (1). For both the position and orientation, we compare the estimation accuracy after the update step. Since the position (and thus the measurement) is influenced by the previous orientation, it contains information on it. However, no information on the change from the previous to the current time step can be obtained. Hence, the accuracy of the orientation estimate is inherently limited by the uncertainty in the system evolution from the previous to the current time step. Therefore, not even a noise-free position measurement could lead to an elimination of all uncertainty in the orientation component.

We will also consider how often the filters failed during the evaluation. The PF can fail when all particle weights are zero after an update step. Since the likelihood is nonzero everywhere, this can only happen due to numerical imprecision (double precision was used). The S3F could potentially fail if all grid points are 0 in some time step.

Based on these criteria, we performed two evaluations. In the first, we extracted the estimates in all time steps and considered the error over the course of the time steps. While the filter results were highly variable in a single run, we also determined the averages over multiple runs to see whether the filter performance degrades over time or stays constant. This evaluation also showed us when the effect of the prior density had mostly vanished for the tracker. We chose to use only 100 particles for the PF and 15 areas for the S3F for this evaluation to see if the filters tend to diverge for low numbers of particles or areas.

In the second evaluation, we aimed to assess the accuracy of the filters during longer tracking tasks. For this, we chose to only consider the accuracy at the 50th time step. Our first evaluation showed that the filters had reached a stable accuracy by then. We did not average over the errors in all times steps to eliminate the effect of the first time steps, in which the prior density still influences the filter result or the filter has not yet reached a stable accuracy. Further, averaging over the errors in all time steps would have made us average over correlated errors. The run times are only considered in the second evaluation. We did not extract an estimate in every time step in this evaluation, and thus, the run times measured exclude the extraction of an estimate.

4.4. Evaluation Results

In the first subsection, we provide the results of our first evaluation, in which we only considered specific configurations for all filters. The results of the second evaluation, in which we considered different numbers of particles for the PF and areas for the S3F, are given in the second subsection.

4.4.1. Evaluation Considering All Time Steps

We first consider a single run and depict the position errors in Figure 6 and the orientation errors in Figure 7. We can see that the error goes up and down over time, which is what we had expected. While most filters work well on average, the system can evolve in a much different way than we expect it to do on average due to the effects of the random

variables in the system model. Furthermore, with the considered Gaussian measurement noise, the obtained measurements in each time step can have arbitrarily large errors, just with small errors being more likely than large errors. From the results for a single run, it is not evident which filter performs best. It is merely evident that the PF with 100 particles once goes off track in its position estimate and appears to be worse overall.

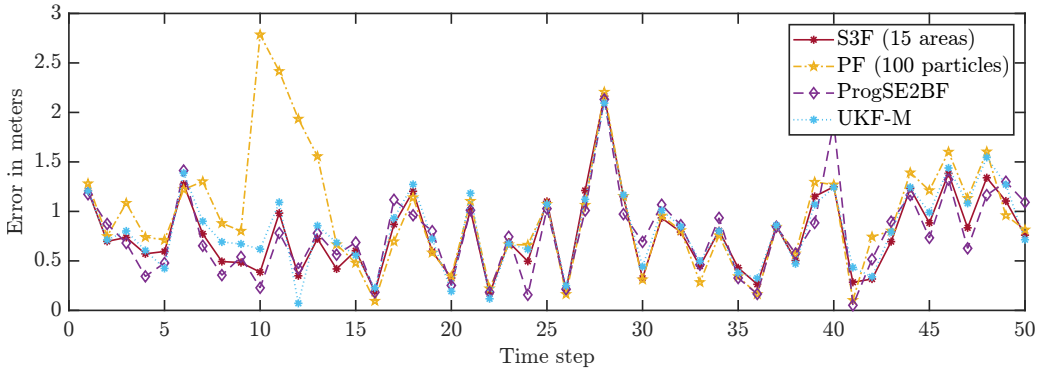

Figure 6. Position errors in all time steps in an example run.

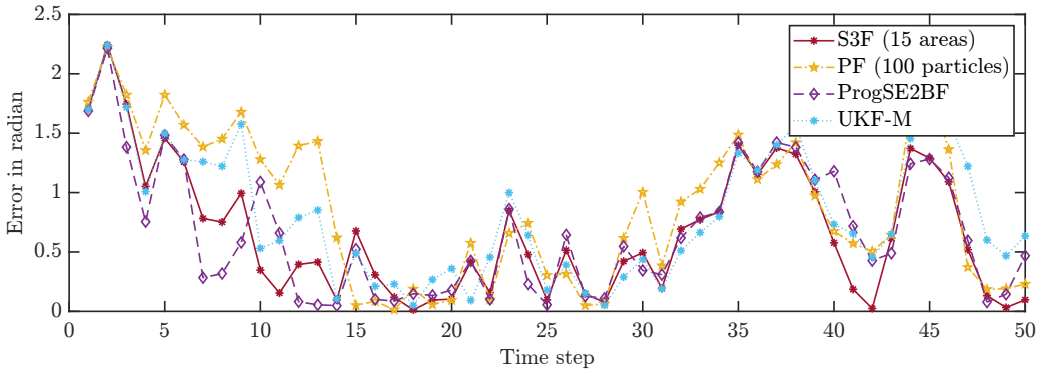

Figure 7. Orientation errors in all time steps in an example run.

To allow for a better comparison of the filters, we averaged the errors in all time steps over 200,000 runs. The results are shown in Figures 8 and 9. None of the considered filters failed in any of the runs. We can see that the S3F with 15 areas performed best both concerning the position and the orientation estimates. The UKF-M yielded better position estimates than the ProgSE2BF. However, the ProgSE2BF provided better orientation estimates. The PF only outperformed the UKF-M in the beginning and was the worst filter at the end of the run. We can see that the UKF-M is slow to converge to a stable quality. When it does, it performs better than the PF with 100 particles. However, as we will see in the next section, the PF can provide better estimates than both the UKF-M and the ProgSE2BF given a sufficient number of samples. Due to the high number of runs, the oscillating behavior of the UKF-M in the error in the orientation component is certain not to be purely noise and may be attributed to its suboptimal handling of the information on the orientation. An important observation for the second evaluation is that it is fair to compare the filters at the 50th time step if one wants to analyze the long-term accuracy of the filters.

4.4.2. Evaluation of the Last Estimates and Run Times

The filters were evaluated based on 70,000 runs in this evaluation. Several configurations were considered for the PF and S3F. The PF with 5 particles failed in 7014 runs and the PF with 11 particles in 145 runs. No failures were observed for all the other configurations and filters.

In Figures 10 and 11, we show the average errors over the number of areas or particles. The ProgSE2BF and UKF-M have a fixed number of parameters and are therefore shown as straight lines. Using only 15 areas, the S3F achieved a higher mean accuracy than a PF with

1,000,000 particles. Using 7 areas, the S3F achieved better accuracy than the ProgSE2BF for the position and orientation. The UKF-M is better than the S3F with 3 areas but worse than the S3F with 7 areas. The PF achieves a better accuracy than the ProgSE2BF when using 700 particles. Using 15 areas, the S3F already achieved very high accuracy. The difference to the accuracy obtained using 30 areas is only in the order of 10^{-6}. The accuracy of the S3F with 15 areas is also higher (although only slightly) than that of a particle filter using 1,000,000 particles. We believe this accuracy to be close to the accuracy an optimal Bayes filter would achieve in this scenario.

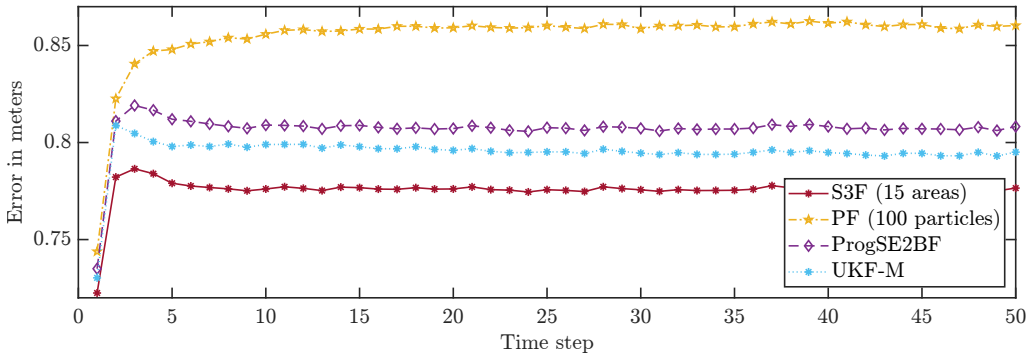

Figure 8. Average position errors in all time steps.

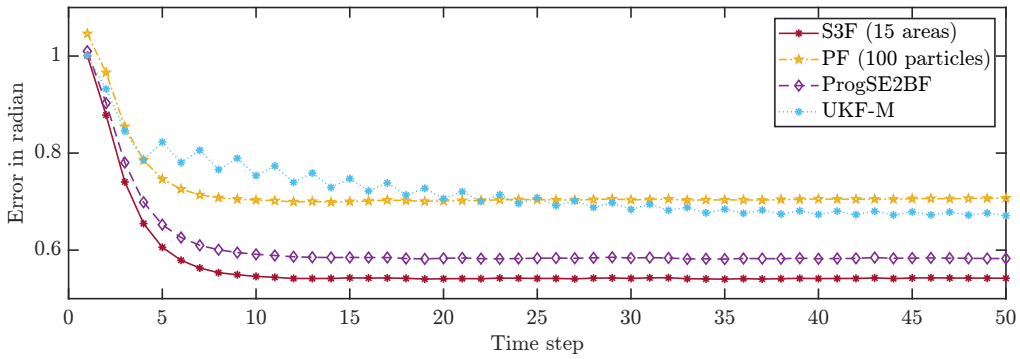

Figure 9. Average orientation errors in all time steps.

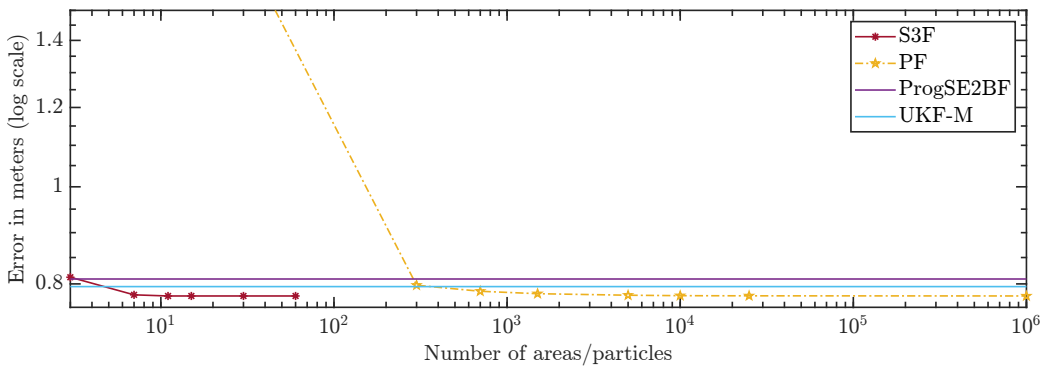

Figure 10. Position error over number of areas or particles.

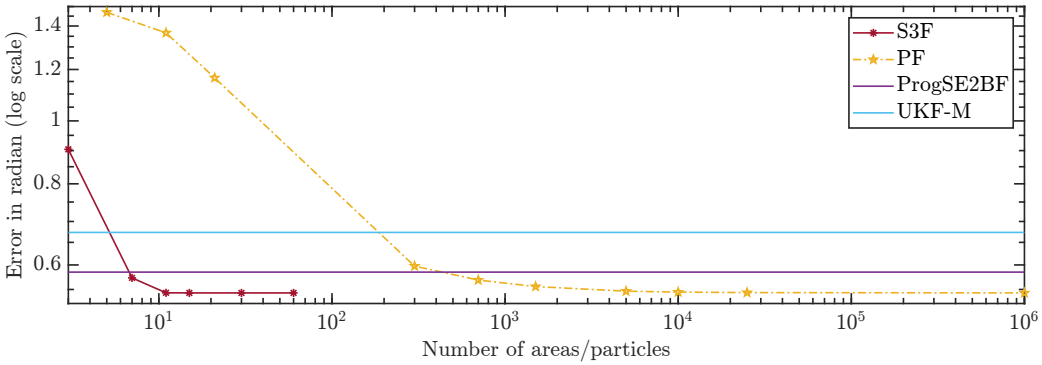

Figure 11. Orientation error over number of areas or particles.

The run times of the filters are shown in Figure 12. We can see that the S3F is much slower per area than the PF per particle, which was expected. We also observe an expected faster increase in the run time for the S3F. While the PF is in $O(n)$, the S3F is in $O(n^2)$. The S3F is faster than the ProgSE2BF in all considered configurations. The PF with 25,000 particles had a run time similar to that of the ProgSE2BF but required much more run time with 1,000,000 particles. The UKF-M was faster than all other filters except the PF with 21 or fewer particles.

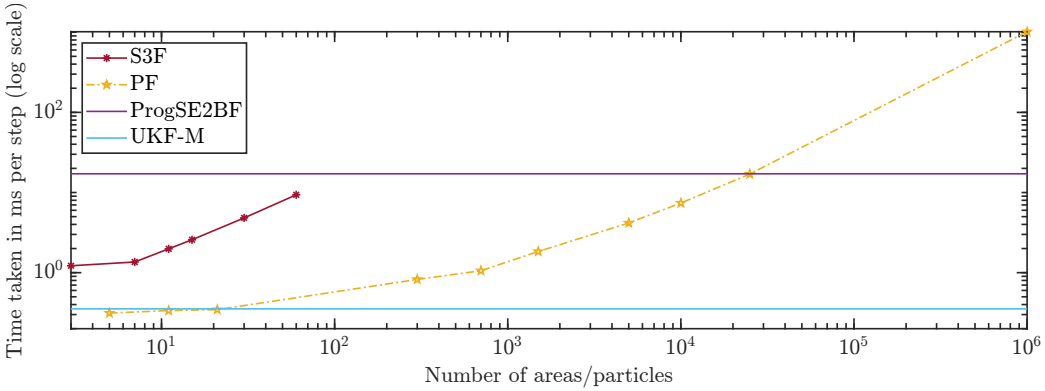

Figure 12. Run time over number of areas or particles.

For a fair comparison of the filters, which have very different run times, we compare the errors over the run times in Figures 13 and 14. A configuration to the lower left of another is strictly better because more accurate results were achieved using less run time. Configurations to the upper left were faster but provided less accurate results. Ones to the lower right were slower but better in terms of accuracy. The latter two cases do not allow for a clear ranking. The plots can be used to assess which filters should be considered given certain run time constraints. However, they can only serve as a rough guideline since the run times highly depend on the scenario, implementation, and hardware.

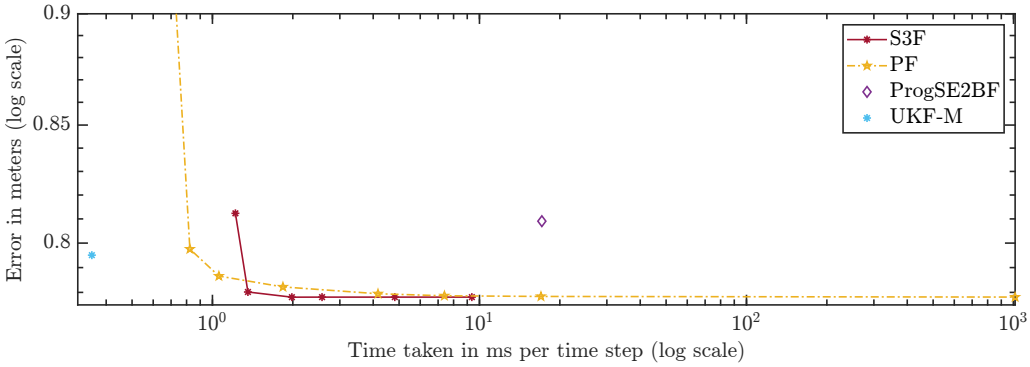

Figure 13. Position error over run time.

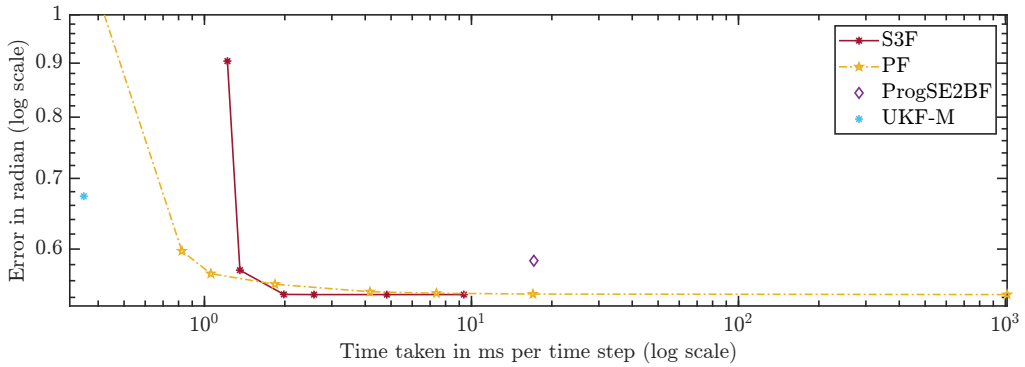

Figure 14. Orientation error over run time.

The S3F using 3 areas is worse than the PF with 300 and 700 particles. However, for configurations of the PF with 5000 particles and more, one can find at least one configuration of the S3F that performs better. In the configuration using 15 areas, the S3F still uses less than 3 ms per time step and is faster and more accurate than the PF with 1,000,000 particles.

Using 7 areas, the S3F achieves better accuracy than the ProgSE2BF at lower run times. The ProgSE2BF is slower than the UKF-M and provides worse position estimates, but its orientation estimates are better, and thus, none of the two filters is clearly superior to the other. The UKF-M is better than the S3F using 3 areas. However, using 7 areas, the S3F yields more accurate results than the UKF-M while taking only less than 1.4 ms per time step. The PF provides more accurate results than the UKF-M in the configuration with 700 particles, in which it takes less than 1.1 ms per time step.

5. Discussion, Summary, and Outlook

In the first subsection, we compare our novel filter with other state-of-the-art approaches based on their theoretical properties and evaluation results. The conclusion of our paper is provided in the second subsection. The third subsection concludes the paper with an outlook.

5.1. Discussion

While some existing approaches, such as the UKF-M and the ProgSE2BF, use a fixed number of parameters, the number of parameters can be increased to arbitrary numbers in the S3F and the PF. Generally, a higher number of particles in the PF or areas in the S3F leads to better results.

If good results at rapid run times are desired or only very low computation power is available, the UKF-M may be a good choice. The S3F should be preferred over the ProgSE2BF as the S3F can achieve better results at lower run times. Generally, both the PF and S3F can achieve higher accuracy than the UKF-M and the ProgSE2BF. Thus, if sufficient computation power is available, one should rather employ the S3F or PF instead of the UKF-M or ProgSE2BF.

While the run time of the PF only increases linearly in the number of particles, the accuracy of its estimates may converge only slowly. This is also reflected in the evaluation results in Section 4.4. The increase in the run time for the S3F is linear in the update step and quadratic in the prediction step, leading to a total complexity of $O(n^2)$. However, the convergence of the accuracy is so much faster than for the PF, and almost optimal results were achieved with relatively low run times. Hence, we do not recommend using a PF with a high number of particles but rather an S3F. Another advantage of the S3F is that it is deterministic, while the PF is non-deterministic (unless one uses a fixed seed for the random number generator).

In our evaluation, good results were obtained, even though none of the considered densities fulfilled the assumptions (A1)–(A3) and (B2)–(B4). However, higher numbers of areas may be required for high-quality estimates when the densities are more concentrated.

Different ways to provide a continuous density exist for the S3F. No densities are inherently provided by the PF. A limitation of the S3F is that one has to provide a transition density for the orientation that is conditionally independent of the previous state. If no such transition density can be provided and no good approximation can be generated using information from the filter, one may consider using the PF instead.

5.2. Summary

In this paper, we presented a novel filter for $\mathrm{SE}(2)$ for which we rewrote the joint density for the full $\mathrm{SE}(2)$ state as a marginalized density for the orientation and a conditional density for the position. By subdividing the domain along the periodic dimension into multiple areas, we were able to present an amalgamation of a grid filter and a Kalman filter. While the S3F for $\mathrm{SE}(2)$ states requires more operations than a 1-D grid filter for $\mathbb{S}$ that permits the use of arbitrary transition densities [20], it is also in $O(n^2)$ and thus in the same complexity class.

Our filter does not support all kinds of system models in the prediction step. Further, since densities do generally not fulfill the assumptions introduced in this paper, the filter involves approximations. However, in our evaluation, the S3F showed a much better convergence than the PF and achieved a very high accuracy using a low number of areas. Using 15 areas, higher accuracy was achieved than with a PF with 1,000,000 particles. The S3F also outperformed the ProgSE2BF by achieving a slightly higher accuracy using considerably less run time. While not as fast as the UKF-M, the S3F can achieve far higher accuracy.

5.3. Outlook

Future work will involve applying the S3F to $\mathrm{SE}(3)$ [32] and other manifolds that can be written as a Cartesian product of periodic (or bounded) and linear manifolds. As another possible extension, using ideas of the extended or unscented Kalman filter in the prediction step may help to allow for a certain degree of nonlinearity in the system model for the position.

Throughout this paper, we assumed the system and measurement models to be precisely known. Future work may concern approaches for imprecisely known models. One way to deal with imprecision in the parameters of linear models on linear domains is to employ robust filters [33,34]. For $\mathrm{SE}(2)$, even if, e.g., only the measurement model for the position is imprecise, this will have to be also considered for the orientation as the position and orientation are generally dependent.

Finally, adaptive subdivisions may help to keep the number of areas low without sacrificing too much accuracy. The basis for this could be adaptive grids as used in [35]. Reducing the required number of areas is particularly important for higher dimensions, e.g., for estimating the pose of a drone along with sensor drifts, biases, and other relevant quantities.

Author Contributions: F.P. proposed and developed the novel filter. K.L. engaged in helpful technical discussions. U.D.H. supervised the work. F.P. prepared the manuscript, and all the authors contributed to polishing it. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially funded by the Helmholtz AI Cooperation Unit within the scope of the project Ubiquitous Spatio-Temporal Learning for Future Mobility (ULearn4Mobility). The authors acknowledge support by the state of Baden-Württemberg through bwHPC. They also acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data for all evaluation runs were generated artificially. The code to generate data and links to the data used for this paper are available at https://github.com/KIT-ISAS/Sensors21_S3FforSE2 (accessed on 19 August 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tayefi, M.; Geng, Z.; Peng, X. Coordinated Tracking for Multiple Nonholonomic Vehicles on SE(2). *Nonlinear Dyn.* **2017**, *87*, 665–675. [CrossRef]
2. Triggs, B. Motion Planning for Nonholonomic Vehicles: An Introduction. 2010. inria-00548415. Available online: https://hal.inria.fr/inria-00548415 (accessed on 17 August 2021).
3. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-Time Loop Closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
4. Chirikjian, G.S. *Stochastic Models, Information Theory, and Lie Groups, Volume 2*; Birkhäuser Boston: Boston, MA, USA, 2012.
5. Adams, R.M.; Biggs, R.; Remsing, C.C. Two-Input Control Systems on the Euclidean Group SE(2). *ESAIM Control Optim. Calc. Var.* **2013**, *19*, 947–975. [CrossRef]
6. Xuan, S.; Li, S.; Han, M.; Wan, X.; Xia, G.S. Object Tracking in Satellite Videos by Improved Correlation Filters with Motion Estimations. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 1074–1086. [CrossRef]
7. Li, X.R.; Jilkov, V.P. Survey of Maneuvering Target Tracking. Part I: Dynamic models. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1333–1364.
8. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In Proceedings of the AAAI National Conference on Artificial Intelligence (AAAI 2002), Edmonton, AB, Canada, 28 July–1 August 2002.
9. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [CrossRef]
10. Li, T.; Sun, S.; Sattar, T.P.; Corchado, J.M. Fight Sample Degeneracy and Impoverishment in Particle Filters: A Review of Intelligent Approaches. *Expert Syst. Appl.* **2014**, *41*, 3944–3954. [CrossRef]
11. Kokkala, J.; Särkkä, S. On the (Non-)Convergence of Particle Filters with Gaussian Importance Distributions. In Proceedings of the 17th IFAC Symposium on System Identification (SYSID 2015), Beijing, China, 19–21 October 2015.
12. Barrau, A.; Bonnabel, S. Invariant Kalman Filtering. *Annu. Rev. Control. Robot. Auton. Syst.* **2018**, *1*, 237–257. [CrossRef]
13. Brossard, M.; Barrau, A.; Bonnabel, S. A Code for Unscented Kalman Filtering on Manifolds (UKF-M). In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA 2020), Paris, France, 31 May–31 August 2020; pp. 5701–5708.
14. Li, K.; Kurz, G.; Bernreiter, L.; Hanebeck, U.D. Nonlinear Progressive Filtering for SE(2) Estimation. In Proceedings of the 21st International Conference on Information Fusion (Fusion 2018), ISIF, Cambridge, UK, 10–13 July 2018.
15. Li, K.; Pfaff, F.; Hanebeck, U.D. Dual Quaternion Sample Reduction for SE(2) Estimation. In Proceedings of the 23rd International Conference on Information Fusion (Fusion 2020), Virtual Conference, Sun City, South Africa, 6–9 July 2020.
16. Murphy, K.; Russell, S. Rao–Blackwellized Particle Filtering for Dynamic Bayesian Networks. In *Sequential Monte Carlo Methods in Practice*; Springer: New York, NY, USA, 2001; pp. 499–515.
17. Matoušek, J.; Duník, J.; Straka, O. Density Difference Grid Design in a Point-Mass Filter. *Energies* **2020**, *13*, 4080. [CrossRef]
18. Kurz, G.; Pfaff, F.; Hanebeck, U.D. Application of Discrete Recursive Bayesian Estimation on Intervals and the Unit Circle to Filtering on SE(2). *IEEE Trans. Ind. Inform.* **2018**, *14*, 1197–1206. [CrossRef]
19. Pfaff, F.; Li, K.; Hanebeck, U.D. Fourier Filters, Grid Filters, and the Fourier-Interpreted Grid Filter. In Proceedings of the 22nd International Conference on Information Fusion (Fusion 2019), ISIF, Ottawa, ON, Canada, 2–5 July 2019.
20. Pfaff, F.; Li, K.; Hanebeck, U.D. Estimating Correlated Angles Using the Hypertoroidal Grid Filter. In Proceedings of the 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2020), Karlsruhe, Germany, 14–16 September 2020.
21. Zygmund, A. *Trigonometric Series*, 3rd ed.; Cambridge University Press: Cambridge, UK, 2003; Volumes 1–2.
22. Pfaff, F.; Li, K.; Hanebeck, U.D. The Spherical Grid Filter for Nonlinear Estimation on the Unit Sphere. In Proceedings of the 1st Virtual IFAC World Congress (IFAC-V 2020), Virtual Conference, Berlin, Germany, 11–17 July 2020.
23. Pfaff, F.; Li, K.; Hanebeck, U.D. A Hyperhemispherical Grid Filter for Orientation Estimation. In Proceedings of the 23rd International Conference on Information Fusion (Fusion 2020), Virtual Conference, Rustenburg, South Africa, 6–9 July 2020.
24. Kurz, G.; Gilitschenski, I.; Hanebeck, U.D. The Partially Wrapped Normal Distribution for SE(2) Estimation. In Proceedings of the 2014 IEEE International Conference on Multisensor Fusion and Information Integration (MFI 2014), Beijing, China, 28–29 September 2014.
25. Mardia, K.V.; Jupp, P.E. *Directional Statistics*; John Wiley & Sons: Chichester, UK, 2000.
26. Mahler, R.P.S. *Statistical Multisource-Multitarget Information Fusion*; Artech House, Inc.: Norwood, MA, USA, 2007.
27. Crouse, D.F.; Willett, P.; Pattipati, K.; Svensson, L. A Look at Gaussian Mixture Reduction Algorithms. In Proceedings of the 14th International Conference on Information Fusion (Fusion 2011), Chicago, IL, USA, 5–8 July 2011; pp. 1–8.
28. Kurz, G.; Gilitschenski, I.; Pfaff, F.; Drude, L.; Hanebeck, U.D.; Haeb-Umbach, R.; Siegwart, R.Y. Directional Statistics and Filtering Using libDirectional. *arXiv* **2017**, arXiv:1712.09718.
29. Brossard, M.; Bonnabel, S.; Condomines, J.P. Unscented Kalman Filtering on Lie Groups. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2017), Vancouver, BC, Canada, 24–28 September 2017.
30. Julier, S.J.; Uhlmann, J.K. Unscented Filtering and Nonlinear Estimation. *Proc. IEEE* **2004**, *92*, 401–422. [CrossRef]

31. Gilitschenski, I.; Kurz, G.; Julier, S.J.; Hanebeck, U.D. A New Probability Distribution for Simultaneous Representation of Uncertain Position and Orientation. In Proceedings of the 17th International Conference on Information Fusion (Fusion 2014), Salamanca, Spain, 7–10 July 2014.
32. Wang, Y.; Chirikjian, G.S. Nonparametric Second-order Theory of Error Propagation on Motion Groups. *Int. J. Robot. Res.* **2008**, *27*, 1258–1273. [CrossRef] [PubMed]
33. Xie, L.; Soh, Y.C. Robust Kalman Filtering for Uncertain Systems. *Syst. Control Lett.* **1994**, *22*, 123–129. [CrossRef]
34. Moheimani, S.; Savkin, A.; Petersen, I. Robust Filtering, Prediction, Smoothing, and Observability of Uncertain Systems. *IEEE Trans. Circuits Syst. Fundam. Theory Appl.* **1998**, *45*, 446–457. [CrossRef]
35. Li, K.; Pfaff, F.; Hanebeck, U.D. Grid-Based Quaternion Filter for SO(3) Estimation. In Proceedings of the 2020 European Control Conference (ECC 2020), St. Petersburg, Russia, 12–15 May 2020.