The State Space Subdivision Filter for SE(3)

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

***Abstract*—Estimating the position and orientation of 3-D objects is a ubiquitous challenge. In our novel filter, the position and orientation of objects are modeled using the Cartesian product of $\mathbb{R}^3$ for the position and a 3-D hyperhemisphere. The latter is used to describe orientations in the form of unit quaternions. The hyperhemisphere is subdivided into equally sized areas. The joint density for the position and orientation is split up into a marginal density for the orientation and a density for the position that is conditioned on the orientation. In our filter, we assume that the function values of the marginal density and the conditional density is the same for all points within that area. By assuming all conditional densities to be Gaussians, efficient formulae can be implemented for the update and prediction steps. The filter is evaluated based on a simulation scenario, for which it showed very high accuracy at low run times.**

***Index Terms*—Grid filter; nonlinear filtering; periodic manifold; special Euclidean group**

I. Introduction

Tracking the pose objects in 3-D is a ubiquitous problem in engineering. A full description of an object's pose comprises its position and orientation. Poses can be seen as elements of the special Euclidean group $\text{SE}(3)$ [1]. For practical algorithms, different representations of poses are used. While the position is generally described by a vector in $\mathbb{R}^3$, there are multiple common ways to represent 3-D orientations. There are also representations that do not inherently separate the position from the orientation, such as unit dual quaternions [2].

A variety of different approaches to tracking on $\text{SE}(3)$ have been employed in the literature. One approach is to use an extended or unscented Kalman filter [3] and hence to assume that the space is locally linear. Generally, the error state extended Kalman filter [4] and the error state unscented Kalman filter [5] are preferred over the regular extended and unscented Kalman filter. In [6], it was proposed to use multiple measurements of specific forms to obtain a linear measurement model. Another approach is to consider the Lie algebra of $\text{SE}(3)$ [1], [7]. Particularly for inertial navigation, the recently introduced Lie group $\text{SE}_2(3)$ [8], [9] has also gained interest.

For $\text{SE}(3)$, it is possible to employ the invariant extended Kalman filter [10], [11], the invariant unscented Kalman filter [12], and the unscented Kalman filter for manifolds [13]. Moreover, it is not difficult to adapt a particle filter (PF) [14] to the domain. Out of the approaches mentioned, only the PF allows increasing the accuracy of the estimates by increasing the number of parameters. However, the convergence of the PF is slow and the PF does not provide a continuous density. The approach we propose in this paper provides a continuous density and its accuracy can be increased by increasing the number of parameters.

Recently, the state space subdivision filter (S3F) [15] was proposed for estimating planar poses, which can be seen as elements of $\text{SE}(2)$. Planar poses can be described by a 2-D translatory part $\underline{x}^\tau$ and an orientation x^ω, which is merely an angle in 2-D. The foundation of the S3F is to rewrite the joint density for the position and orientation $f(\underline{x}^\tau, x^\omega)$ as the product of the marginalized density for the periodic part $f^\omega(x^\omega)$ and the conditional density $f^\text{c}(\underline{x}^\tau|x^\omega)$. Then, the circular manifold is subdivided into n equal-sized areas. The marginalized density is represented using function values on a grid, as is done in the grid filter for the unit circle presented in [16]. For each area, one stores the function value $\gamma_i = f^\omega(\beta_i)$, and the marginal density is assumed to be precisely γ_i in that area. The conditional densities are assumed to be Gaussian. The conditional density can be different for each of the areas. For each area, the mean vector $\underline{\mu}_i$ and covariance matrix $\mathbf{C}_i$ of the conditional density are stored. Using some additional assumptions, which we get into detail later in this paper, prediction and update steps can be performed.

In this paper, we present the S3F for $\text{SE}(3)$. We parameterize $\text{SE}(3)$ states using a 3-D translatory part $\underline{x}^\tau$ and an orientation $\underline{x}^\omega$ in the form of a unit quaternion. Unit quaternions lie on the 3-D unit hypersphere $\mathbb{S}^3$. However, since the quaternions $\underline{q}$ and $-\underline{q}$ describe same orientation, it is sufficient to consider the upper (along the last dimension) hyperhemisphere of the unit hypersphere, which we denote by $\mathbb{H}^3$. Considering only the upper hemisphere avoids redundancies and prevents asymmetries, which could be caused by approximation errors or numerical imprecision. In our parameterization, the full state vector comprising $\underline{x}^\tau$ and $\underline{x}^\omega$ is an element of $\mathbb{H}^3 \times \mathbb{R}^3$.

We use a grid-based filter for hyperhemispheres [17] as the basis for the S3F for $\text{SE}(3)$. In this filter, $\mathbb{H}^3$ is partitioned into a set of n areas $\mathcal{A} = \{A_1, \dots, A_n\}$. To partition the domain, we presented a symmetrized version of equal area partitioning algorithm [18] in [17]. By considering only the areas on the upper hemisphere, a partition of $\mathbb{H}^3$ can be obtained. The subdivision of $\mathbb{H}^3 \times \mathbb{R}^3$ would thus be $\{A_i \times \mathbb{R}^3 | i \in \{1, \dots, n\}\}$. However, the focus of our considerations will be on the elements in $\mathcal{A}$, and we will use the term area to refer to an element $A_i \subset \mathbb{H}^3$.

In the next section, we explain the assumptions that underlie our filter and how a given density can be approximated in the required parametric form. The formulae for the update and prediction steps are derived in Sec. III. An evaluation is provided in fourth section. We provide a conclusion and an outlook in Sec. V.

II. Assumptions and Density Representation

We begin by formally defining the domain and explaining the relationship between the areas, grid points, and grid values. Then, we detail the assumption that underlie our filter in the first subsection of this section. The second subsection addresses how existing densities are approximated.

If one defined $\mathbb{H}^3$ as the points in $\mathbb{R}^4$ with norm 1 and $x_4 > 0$, it would not contain certain points, e.g., $\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^\top$. If one used the condition $x_4 \geq 0$, $\mathbb{H}^3$ would contain, e.g., both $\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^\top$ and $\begin{bmatrix}-1 & 0 & 0 & 0\end{bmatrix}^\top$. Therefore, we define

$$\mathbb{H}^3 = \{\underline{x} \in \mathbb{R}^4 : \|\underline{x}\| = 1 \wedge (x_4 > 0 \vee x_4 = 0 \wedge x_3 > 0 \\ \vee x_4 = 0 \wedge x_3 = 0 \wedge x_2 > 0 \vee \underline{x} = \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^\top)\}$$

to ensure that a unit quaternion $\underline{q}$ is in $\mathbb{H}^3$ if and only if $-\underline{q} \notin \mathbb{H}^3$.

The subdivision of $\mathbb{H}^3$ presented in [17] ensures that there is a well-defined center for each area A_i. The centers are used as grid points $\mathcal{B} = \{\underline{\beta}_1, \dots, \underline{\beta}_n\}$. As explained more formally in the first subsection, the grid value γ_i describes the function value of the marginal density f^ω in the corresponding area. Note that while $\mathcal{A}$ and $\mathcal{B}$ are sets, they may only be reordered in the same way, which also requires reordering the entries of the corresponding vector of grid values $\underline{\gamma}$.

A. Assumptions

Our filter is based on assumptions that are used to obtain grid-based representations of densities, likelihoods, and transition densities. We first consider densities and likelihoods. Besides the previously mentioned constantness of f^ω in each area, we assume that the conditional densities are the same for every $\underline{x}^\tau$ in each area and that they are all Gaussians. Formally, our assumptions are as follows.

(A1) The marginal density $f^\omega(\underline{x}^\omega)$ (or likelihood $\mathcal{L}_{\underline{\hat{z}}}$) is constant within each area, i.e., $\forall \underline{x}^\omega \in A_i : f^\omega(\underline{x}^\tau) = \gamma_i$ (or $\forall \underline{\alpha} \in A_i : \mathcal{L}_{\underline{\hat{z}}}(\underline{x}^\tau) = \gamma_i$).

(A2) The conditional density $f^{\mathrm{c}}(\underline{x}^\tau | \underline{x}^\omega)$ (or likelihood $\mathcal{L}^{\mathrm{c}}_{\underline{\hat{z}}}(\underline{x}^\tau | \underline{x}^\omega)$) is the same for all $\underline{x}^\omega \in A_i$.

(A3) $f^{\mathrm{c}}(\underline{x}^\tau | \underline{x}^\omega)$ is Gaussian for every considered $\underline{x}^\omega$.

By combining (A2) with (A3), we know that the conditional densities for all $\underline{x}^\omega \in A_i$, which we denote by $f^{\mathrm{c}}_i(\underline{x}^\tau)$, can be parameterized by a single mean vector $\underline{\mu}_i$ and covariance matrix $\mathbf{C}_i$. The entire density or likelihood can be described by the n-tupel of triples $\left((\gamma_i, \underline{\mu}_i, \mathbf{C}_i)\right)_{i=1}^n$.

For the prediction step, we need a suitable parameterization for the transition density, which can be written as

$$f^{\mathrm{T}}(\underline{x}^\tau_{t+1}, \underline{x}^\omega_{t+1} | \underline{x}^\tau_t, \underline{x}^\omega_t) \\ = f^{\mathrm{T,c,full}}(\underline{x}^\tau_{t+1} | \underline{x}^\omega_{t+1}, \underline{x}^\tau_t, \underline{x}^\omega_t) f^{\mathrm{T},\omega,\mathrm{full}}(\underline{x}^\omega_{t+1} | \underline{x}^\tau_t, \underline{x}^\omega_t) \ .$$

We introduce a different set of assumptions for the transition density. The first is that the next orientation must not depend on the current translation. In other words, there is a density $f^{\mathrm{T},\omega}(\underline{x}^\omega_{t+1} | \underline{x}^\omega_t)$ that does not depend on $\underline{x}^\tau_t$ but returns the same function values as $f^{\mathrm{T},\omega,\mathrm{full}}(\underline{x}^\omega_{t+1} | \underline{x}^\tau_t, \underline{x}^\omega_t)$. For our discretization-based approximation, we see the transition density as a function on $\mathbb{R}^3 \times \mathbb{H}^3 \times \mathbb{R}^3 \times \mathbb{H}^3$ and discretize $\mathbb{H}^3 \times \mathbb{H}^3$. While other subdivisions would be possible, we shall always use $\mathcal{A} \times \mathcal{A}$ as the subdivision of $\mathbb{H}^3 \times \mathbb{H}^3$. Our grid values are then $\mathcal{B} \times \mathcal{B}$. The grid values describe the function values of $f^{\mathrm{T},\omega}(\underline{x}^\omega_{t+1} | \underline{x}^\omega_t)$ in the areas $\mathcal{A} \times \mathcal{A}$ and are stored in a matrix $\mathbf{\Gamma}^{\mathrm{T}}$. For each of the grid points, one can have a different linear model with white Gaussian noise that is parameterized by a transition matrix $\mathbf{F}_{t,i,j}$, an input vector $\underline{\hat{u}}_{t,i,j}$, and a system noise covariance matrix $\mathbf{C}^{\underline{\boldsymbol{w}}}_{t,i,j}$. Note that the transition matrix, the input vector, and the covariance matrix may depend (nonlinearly) on the regions for the orientation at time step t and $t+1$. They may also only depend on the current orientation $\underline{x}^\omega_t$.

Formally, we obtain the following four assumptions.

(B1) $\forall \underline{x}^\tau_t \in \mathbb{R}^3 : f^{\mathrm{T},\omega,\mathrm{full}}(\underline{x}^\omega_{t+1} | \underline{x}^\tau_t, \underline{x}^\omega_t) = f^{\mathrm{T},\omega}(\underline{x}^\omega_{t+1} | \underline{x}^\omega_t)$.

(B2) $\forall \underline{x}^\omega_{t+1} \in A_i \forall \underline{x}^\omega_t \in A_j :$

$$f^{\mathrm{T},\omega}(\underline{x}^\omega_{t+1} | \underline{x}^\omega_t) = f^{\mathrm{T},\omega}(\beta_i | \beta_j) = \gamma^{\mathrm{T}}_{i,j} \ .$$

(B3) The conditional density $f^{\mathrm{T,c,full}}(\underline{x}^\tau_{t+1} | \underline{x}^\omega_{t+1}, \underline{x}^\tau_t, \underline{x}^\omega_t)$ is the same function for all $\underline{x}^\omega_{t+1} \in A_i$ and $\underline{x}^\omega_t \in A_j$. We denote it by $f^{\mathrm{T,c}}_{i,j}(\underline{x}^\tau_{t+1} | \underline{x}^\tau_t)$.

(B4) Each conditional density can be written as

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

with f_{N} being a Gaussian density with the specified parameters.

Throughout this paper, we assume that one can write the transition density in such a form. In our implementation, we set $f^{\mathrm{T,c}}_{i,j}(\underline{x}^\tau_{t+1} | \underline{x}^\tau_t)$ to $f^{\mathrm{T,c,full}}(\underline{x}^\tau_{t+1} | \beta_i, \underline{x}^\tau_t, \beta_j)$, i.e., we use the conditional density that is obtained by setting $\underline{x}^\omega_{t+1}$ to β_i and $\underline{x}^\omega_t$ to β_j.

B. Density Approximation

Our filter uses parametric densities and likelihoods with the assumptions explained in the previous subsection. We now consider how to obtain parameters for given densities and likelihoods. We put our focus on (normalized) densities. The process can be applied similarly for likelihoods, and we briefly highlight some differences. For the system model, we assume we are given it a suitable parametric form. Note that all intermediate results of the filter are given in parametric form, and thus, approximating a given density is only necessary as part of the initialization or if the likelihood must be approximated anew in each time step. We start by explaining how the grid values can be obtained. Then, we address how we can get the parameters of the conditional densities. Finally, we explain how we can ensure the normalization, which is only relevant for densities and not for likelihoods.

If the density or likelihood is given in the form of a product of a marginal density and a conditional density, the grid values can be obtained directly from the marginal density. If we assume the marginal density to be constant (as assumed in A1), we can simply evaluate the marginal density at the grid points. If we only have the joint density, we can compute the grid values according to

$$\breve{\gamma}_i = f^\omega(\underline{\beta}_i) = \int_{\mathbb{R}^3} f(\underline{x}^\tau, \underline{\beta}_i) d\underline{x}^\tau \ .$$

The original marginal density is not necessarily constant in each of the areas. In this case, the normalization of the marginal density represented by the grid values may not be ensured, which we indicate by the use of the ˘ decorator in our paper. As an alternative, one could integrate the marginal density f^ω over A_i (or f over $A_i \times \mathbb{R}^3$) and divide the result by the size of A_i. While this approach may be beneficial if the marginal density is highly variable within the area, it involves calculating high-dimensional integrals. Therefore, we do not employ this approach for our practical implementation. Instead, we normalize the density as explained at the end of this subsection.

If the conditional densities are not directly given, we need to obtain them from the joint density. We always use the conditional density at respective the grid point, i.e., $f^{\mathrm{c}}(\underline{x}^\tau|\underline{\beta}_i)$ for the conditional density for each area. When the grid values are chosen to be the function values at the grid points (and not the average function value) the conditional density is given according to

$$f_i^{\mathrm{c}}(\underline{x}^\tau) = f^{\mathrm{c}}(\underline{x}^\tau|\underline{\beta}_i) = \frac{f(\underline{x}^\tau, \underline{\beta}_i)}{f^\omega(\underline{\beta}_i)} = \frac{1}{\breve{\gamma}_i} f(\underline{x}^\tau, \underline{\beta}_i) \ .$$

If the conditional densities are not Gaussians, we approximate them with Gaussians by calculating the mean and covariance. If no analytic way to obtain the mean and covariance is available, one can employ the well-known formulae

$$\underline{\mu}_i = \int_{\mathbb{R}^3} \underline{x}^\tau f_i^{\mathrm{c}}(\underline{x}^\tau) d\underline{x}^\tau$$

for the mean vector and

$$c_{i,j,k} = \int_{\mathbb{R}^3} (x_j^\tau - \mu_{i,j})(x_k^\tau - \mu_{i,k}) f_i^{\mathrm{c}}(\underline{x}^\tau) d\underline{x}^\tau$$

for the entry in column j and row k of the covariance matrix for the conditional density for area i. For n areas, n 3-D integrals are required to determine the grid values, and $3n$ and $6n$ integrals are required for the mean vectors and covariance matrices, respectively. While the approximation of the prior density can be done offline and the low run times are thus not vital, the likelihood may need to be approximated during run time. Therefore, the filter is most efficient when the parameters of the likelihood can be obtained analytically.

As mentioned in the beginning of this subsection, the marginal density represented by the grid values may not be normalized when one uses the function values at the grid points as the grid values. Therefore, the joint density may not be normalized, even though the conditional densities are inherently normalized because they are Gaussians. This can be seen via

$$\begin{aligned}
&\int_{\mathbb{R}^3} \int_{\mathbb{H}^3} f\left(\underline{x}^\tau, \underline{x}^\omega; \left((\gamma_i, \underline{\mu}_i, \mathbf{C}_i)\right)_{i=1}^n\right) d\underline{x}^\omega d\underline{x}^\tau \\
&= \sum_{i=1}^n \int_{A_i} f^\omega(\underline{x}^\omega) \int_{\mathbb{R}^3} f_i^{\mathrm{c}}(\underline{x}^\tau) d\underline{x}^\tau d\underline{x}^\omega \\
&= \sum_{i=1}^n \breve{\gamma}_i \left|A_i\right| = \frac{\pi^2}{n} \sum_{i=1}^n \breve{\gamma}_i = \frac{\pi^2}{n} \left\|\breve{\underline{\gamma}}\right\|_1 \ ,
\end{aligned}$$

with $\left\|\breve{\underline{\gamma}}\right\|_1$ being the sum of all entries of the vector $\breve{\underline{\gamma}}$ and $\left|A_i\right|$ being the size of A_i, which is $\frac{\pi^2}{n}$ for equal-sized areas because $\left|\mathbb{H}^3\right| = \pi^2$. From the formula of the integral, we can also see that scaling all values in the vector of grid values by a constant leads to a function whose integral is scaled by that constant. Therefore, we can obtain the vector $\underline{\gamma}$ describing the normalized density by dividing all entries of $\breve{\underline{\gamma}}$ by $\frac{\pi^2}{n}\left\|\breve{\underline{\gamma}}\right\|_1$. Note that likelihood functions need not be normalized. As gets evident in Sec. III-A, our filter works even if the integral of the approximation of the likelihood over the entire domain is not equivalent to the integral of the original likelihood.

III. Filter Derivation

In our derivation, we proceed similarly as in the derivation of the S3F for SE(2) [15]. For the derivation, we require the fundamental Gaussian identity

$$\begin{aligned}
&f_{\mathrm{N}}(\mathbf{M}\underline{x}; \underline{\mu}^r, \mathbf{C}^r) f_{\mathrm{N}}(\underline{x}; \underline{\mu}^s, \mathbf{C}^s) \\
&= f_{\mathrm{N}}(\underline{\mu}^r; \mathbf{M}\underline{\mu}^s, \mathbf{C}^r + \mathbf{M}\mathbf{C}^s\mathbf{M}^\top) f_{\mathrm{N}}(\underline{x}; \underline{\mu}^q, \mathbf{C}^q)
\end{aligned} \tag{2}$$

with

$$\begin{aligned}
\mathbf{C}^q &= \left(\mathbf{M}^\top \left(\mathbf{C}^r\right)^{-1} \mathbf{M} + \left(\mathbf{C}^s\right)^{-1}\right)^{-1} \ , \\
\underline{\mu}^q &= \mathbf{C}^q \left(\mathbf{M}^\top \left(\mathbf{C}^r\right)^{-1} \underline{\mu}^r + \left(\mathbf{C}^s\right)^{-1} \underline{\mu}^s\right) \ .
\end{aligned}$$

The update step is derived in the first subsection and the prediction step in the second subsection.

A. Update Step

In the update step, the measurement $\hat{\underline{z}}_t$ is used according to the likelihood function $\mathcal{L}_{\hat{\underline{z}}_t}$ to improve our knowledge about the current state. We denote the prior density at time step t that only considers all measurements until time step $t-1$ by f_t^{p} and the posterior density also considering $\hat{\underline{z}}_t$ by f_t^{e}. Using Bayes' formula, we obtain

$$\begin{aligned}
&f_t^{\mathrm{e}}(\underline{x}_t^\tau, \underline{x}_t^\omega | \hat{\underline{z}}_1, \ldots, \hat{\underline{z}}_t) \\
&\propto \mathcal{L}_{\hat{\underline{z}}_t}(\underline{x}_t^\tau, \underline{x}_t^\omega) f_t^{\mathrm{p}}(\underline{x}_t^\tau, \underline{x}_t^\omega | \hat{\underline{z}}_1, \ldots, \hat{\underline{z}}_{t-1}) \ ,
\end{aligned}$$

in which $\propto$ indicates that the two terms only differ by a nonnegative factor. From now on, we denote the product of the likelihood and the prior density by $\breve{f}_t^{\mathrm{e}}$, and we omit the dependencies on the measurements for brevity. We further omit the time indices of the densities because it is always t in the update step. Because $f^{\mathrm{e}} \propto \breve{f}^{\mathrm{e}}$, we can implement Bayes' formula by first determining $\breve{f}^{\mathrm{e}}$ via a multiplication and then normalizing the result.

Based on our assumptions, the parametric prior density before the update step is

$$\begin{aligned}
&f^{\mathrm{p}}\left(\underline{x}_t^\tau, \underline{x}_t^\omega, \left((\gamma_j^{\mathrm{p}}, \underline{\mu}_j^{\mathrm{p}}, \mathbf{C}_j^{\mathrm{p}})\right)_{j=1}^n\right) \\
&= \sum_{j=1}^n \mathbb{1}\left\{\underline{x}_t^\omega \in A_j\right\} \gamma_j^{\mathrm{p}} f_{\mathrm{N}}\left(\underline{x}_t^\tau; \underline{\mu}_j^{\mathrm{p}}, \mathbf{C}_j^{\mathrm{p}}\right) \ ,
\end{aligned} \tag{3}$$

with $\mathbb{1}$ being the indicator function that is 1 when the condition in braces is met and 0 otherwise. The parameters are either obtained from the previous prediction step or by approximating the initial prior density as explained in Sec. II-B. Similarly, the likelihood is given by

$$\begin{aligned}&\mathcal{L}_{\hat{\underline{z}}_t}\left(\underline{x}_t^{\mathcal{T}}, \underline{x}_t^{\omega}; \left((\gamma_i^{\text{L}}; \underline{\mu}_i^{\text{L}}, \mathbf{C}_i^{\text{L}})\right)_{i=1}^{n}\right)\\ &= \sum_{i=1}^{n} \mathbb{1}\left\{\underline{x}_t^{\omega} \in A_i\right\} \gamma_i^{\text{L}} f_{\text{N}}\left(\underline{x}_t^{\mathcal{T}}; \underline{\mu}_i^{\text{L}}, \mathbf{C}_i^{\text{L}}\right) \ .\end{aligned}$$

The unnormalized posterior density is given by the multiplication result, leading to

$$\begin{aligned}\breve{f}^{\text{e}}\left(\underline{x}_t^{\mathcal{T}}, \underline{x}_t^{\omega}\right) = &\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{1}\left\{\underline{x}_t^{\omega} \in A_i\right\} \mathbb{1}\left\{\underline{x}_t^{\omega} \in A_j\right\}\\ &\cdot \gamma_i^{\text{L}} \gamma_j^{\text{p}} f_{\text{N}}\left(\underline{x}_t^{\mathcal{T}}; \underline{\mu}_i^{\text{L}}, \mathbf{C}_i^{\text{L}}\right) f_{\text{N}}\left(\underline{x}_t^{\mathcal{T}}; \underline{\mu}_j^{\text{p}}, \mathbf{C}_j^{\text{p}}\right) \ .\end{aligned}$$

All terms with $i \neq j$ are zero, and thus, we can omit the second sum and set j to i. Then, we further employ the fundamental Gaussian identity (2) to obtain

$$\begin{aligned}\breve{f}^{\text{e}}\left(\underline{x}_t^{\mathcal{T}}, \underline{x}_t^{\omega}\right) = \sum_{i=1}^{n} \mathbb{1}\left\{\underline{x}_t^{\omega} \in A_i\right\}\\ \gamma_i^{\text{L}} \gamma_i^{\text{p}} f_{\text{N}}\left(\underline{\mu}_i^{\text{p}}; \underline{\mu}_i^{\text{L}}, \mathbf{C}_i^{\text{L}} + \mathbf{C}_i^{\text{p}}\right) f_{\text{N}}\left(\underline{x}_t^{\mathcal{T}}; \underline{\mu}_i^{\text{e}}, \mathbf{C}_i^{\text{e}}\right) \ ,\end{aligned} \tag{4}$$

with

$$\mathbf{C}_i^{\text{e}} = \left((\mathbf{C}_i^{\text{L}})^{-1} + (\mathbf{C}_i^{\text{p}})^{-1}\right)^{-1} \ , \tag{5}$$

$$\underline{\mu}_i^{\text{e}} = \mathbf{C}_i^{\text{e}}((\mathbf{C}_i^{\text{L}})^{-1}\underline{\mu}_i^{\text{L}} + (\mathbf{C}_i^{\text{p}})^{-1}\underline{\mu}_i^{\text{p}}) \ . \tag{6}$$

The parameters of the conditional densities are directly given by (5) and (6).

From (4), we obtain the formula

$$\breve{\gamma}_i^{\text{e}} = \gamma_i^{\text{L}} \gamma_i^{\text{p}} f_{\text{N}}\left(\underline{\mu}_i^{\text{p}}; \underline{\mu}_i^{\text{L}}, \mathbf{C}_i^{\text{L}} + \mathbf{C}_i^{\text{p}}\right) f_{\text{N}}\left(\underline{x}_t^{\mathcal{T}}; \underline{\mu}_i^{\text{e}}, \mathbf{C}_i^{\text{e}}\right)$$

for the grid values of the unnormalized posterior density. Then, we determine the vector $\underline{\gamma}^{\text{e}}$ describing the normalized result by dividing $\breve{\underline{\gamma}}^{\text{e}}$ by $\left(\pi^2 \left\|\breve{\underline{\gamma}}^{\text{e}}\right\|_1\right)$ (see Sec. II-B for information on the normalization). For n areas, all operations involved are in $O(n)$.

B. Prediction Step

The prediction step provides the prior density for time step $t+1$ based on the posterior density for time step t, which is obtained from the most recent update step. Our prediction step is based on the Chapman–Kolmogorov equation

$$\begin{aligned}&f_{t+1}^{\text{p}}(\underline{x}_{t+1}^{\mathcal{T}}, \underline{x}_{t+1}^{\omega} | \hat{\underline{z}}_1, \ldots, \hat{\underline{z}}_t)\\ =&\int_{\mathbb{R}^3}\int_{\mathbb{H}^3} \underbrace{f_t^{\text{T}}(\underline{x}_{t+1}^{\mathcal{T}}, \underline{x}_{t+1}^{\omega} | \underline{x}_t^{\mathcal{T}}, \underline{x}_t^{\omega}) f_t^{\text{e}}(\underline{x}_t^{\mathcal{T}}, \underline{x}_t^{\omega} | \hat{\underline{z}}_1, \ldots, \hat{\underline{z}}_t)}_{f_t^{\text{j}}(\underline{x}_{t+1}^{\mathcal{T}}, \underline{x}_{t+1}^{\omega}, \underline{x}_t^{\mathcal{T}}, \underline{x}_t^{\omega})} d\underline{x}_t^{\omega} d\underline{x}_t^{\mathcal{T}} \ ,\end{aligned}$$

in which the system model is given in the form of a transition density f_t^{T} that fulfills Sec. II-A. We implement the prediction step by first multiplying f_t^{T} with f_t^{e}, which yields the joint density $f_t^{\text{j}}(\underline{x}_{t+1}^{\mathcal{T}}, \underline{x}_{t+1}^{\omega}, \underline{x}_t^{\mathcal{T}}, \underline{x}_t^{\omega})$ as an intermediate result. We then marginalize $\underline{x}_t^{\omega}$ and $\underline{x}_t^{\mathcal{T}}$ out to obtain f_{t+1}^{p}.

First, we write out the transition density based on the assumptions (B1)–(B3), leading to

$$\begin{aligned}&f^{\text{T}}\left(\underline{x}_{t+1}^{\mathcal{T}}, \underline{x}_{t+1}^{\omega} | \underline{x}_t^{\mathcal{T}}, \underline{x}_t^{\omega}; \left\{\gamma_{i,j}^{\text{T}}, \mathbf{F}_{i,j}, \hat{\underline{u}}_{i,j}^{\mathcal{T}}, \mathbf{C}_{i,j}^{\boldsymbol{w}}\right\}_{(i,j)\in\{1,\ldots,n\}^2}\right)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{1}\left\{\underline{x}_{t+1}^{\omega} \in A_i\right\} \mathbb{1}\left\{\underline{x}_t^{\omega} \in A_j\right\} \gamma_{i,j}^{\text{T}} f_{i,j}^{\text{T,c}}\left(\underline{x}_{t+1}^{\mathcal{T}} \middle| \underline{x}_t^{\mathcal{T}}\right) .\end{aligned} \tag{7}$$

The joint density is then the product of the double sum for the transition density (7) and the sum for the posterior density. We first multiply out the sum and then reduce the resulting triple sum to a double sum by considering that all terms with $k \neq j$ are zero, yielding

$$\begin{aligned}&f^{\text{j}}(\underline{x}_{t+1}^{\mathcal{T}}, \underline{x}_{t+1}^{\omega}, \underline{x}_t^{\mathcal{T}}, \underline{x}_t^{\omega})\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \mathbb{1}\left\{\underline{x}_{t+1}^{\omega} \in A_i\right\} \mathbb{1}\left\{\underline{x}_t^{\omega} \in A_j\right\} \mathbb{1}\left\{\underline{x}_t^{\omega} \in A_k\right\}\\ &\qquad \cdot \gamma_{i,j}^{\text{T}} \gamma_k^{\text{e}} f_{i,j}^{\text{T,c}}\left(\underline{x}_{t+1}^{\mathcal{T}} \middle| \underline{x}_t^{\mathcal{T}}\right) f_k^{\text{e}}\left(\underline{x}_t^{\mathcal{T}}\right)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{1}\left\{\underline{x}_{t+1}^{\omega} \in A_i\right\} \mathbb{1}\left\{\underline{x}_t^{\omega} \in A_j\right\}\\ &\qquad \cdot \gamma_{i,j}^{\text{T}} \gamma_j^{\text{e}} f_{i,j}^{\text{T,c}}\left(\underline{x}_{t+1}^{\mathcal{T}} \middle| \underline{x}_t^{\mathcal{T}}\right) f_j^{\text{e}}\left(\underline{x}^{\mathcal{T}}\right) \ .\end{aligned}$$

We now define $\gamma_{i,j}^{\text{j}} = \gamma_{i,j}^{\text{T}} \gamma_j^{\text{e}}$ and marginalize out $\underline{x}_t^{\omega}$ and $\underline{x}_t^{\mathcal{T}}$ to obtain the predicted density

$$\begin{aligned}&f_{t+1}^{\text{p}}\left(\underline{x}_{t+1}^{\mathcal{T}}, \underline{x}_{t+1}^{\omega}\right)\\ &= \int_{\mathbb{H}^3}\int_{\mathbb{R}^3}\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{1}\left\{\underline{x}_{t+1}^{\omega} \in A_i\right\} \mathbb{1}\left\{\underline{x}_t^{\omega} \in A_j\right\}\\ &\quad \gamma_{i,j}^{\text{j}} f_{i,j}^{\text{T,c}}\left(\underline{x}_{t+1}^{\mathcal{T}} \middle| \underline{x}_t^{\mathcal{T}}\right) f_j^{\text{e}}\left(\underline{x}_t^{\mathcal{T}}\right) d\underline{x}_t^{\mathcal{T}} d\underline{x}_t^{\omega}\\ &= \sum_{i=1}^{n} \mathbb{1}\left\{\underline{x}_{t+1}^{\omega} \in A_i\right\} \sum_{j=1}^{n} \int_{\mathbb{R}^3} \gamma_{i,j}^{\text{j}}\\ &\quad \underbrace{f_{i,j}^{\text{T,c}}\left(\underline{x}_{t+1}^{\mathcal{T}} \middle| \underline{x}_t^{\mathcal{T}}\right) f_j^{\text{e}}\left(\underline{x}_t^{\mathcal{T}}\right) d\underline{x}^{\mathcal{T}}}_{\zeta} \underbrace{\int_{\mathbb{H}^3} \mathbb{1}\left\{\underline{x}_t^{\omega} \in A_j\right\} d\underline{x}_t^{\omega}}_{\eta} \ .\end{aligned} \tag{8}$$

η is the size of A_j. In our subdivision using equal-sized areas, η is $\frac{\pi^2}{n}$. We now write out the densities, which are assumed to be Gaussian according to (A3) and (B4), and apply the fundamental Gaussian identity to obtain

$$\begin{aligned}\zeta &= f_{\text{N}}\left(\underline{x}_{t+1}^{\mathcal{T}}; \mathbf{F}_{i,j}\underline{x}_t^{\mathcal{T}} + \hat{\underline{u}}_{i,j}^{\mathcal{T}}, \mathbf{C}_{i,j}^{\boldsymbol{w}}\right) f_{\text{N}}\left(\underline{x}_t^{\mathcal{T}}; \underline{\mu}_j^{\text{e}}, \mathbf{C}_j^{\text{e}}\right)\\ &= \left(\mathbf{F}_{i,j}\underline{x}_t^{\mathcal{T}}; \underline{x}_{t+1}^{\mathcal{T}} - \hat{\underline{u}}_{i,j}^{\mathcal{T}}, \mathbf{C}_{i,j}^{\boldsymbol{w}}\right) f_{\text{N}}\left(\underline{x}_t^{\mathcal{T}}; \underline{\mu}_j^{\text{e}}, \mathbf{C}_j^{\text{e}}\right)\\ &= f_{\text{N}}\left(\underline{x}_{t+1}^{\mathcal{T}} - \hat{\underline{u}}_{i,j}^{\mathcal{T}}; \mathbf{F}_{i,j}\underline{\mu}_j^{\text{e}}, \mathbf{C}_{i,j}^{\boldsymbol{w}} + \mathbf{F}_{i,j}\mathbf{C}_j^{\text{e}}\mathbf{F}_{i,j}^{\top}\right)\\ &\quad f_{\text{N}}\left(\underline{x}_t^{\mathcal{T}}; \underline{\mu}^{q}, \mathbf{C}^{q}\right) \ .\end{aligned}$$

The values in $\underline{\mu}^{q}$ and $\mathbf{C}^{q}$ in the final expression are irrelevant. The density of the normal distribution is the only term that depends on $\underline{x}_t^{\mathcal{T}}$ in the integral (8) over the entire domain, which is 1 because the normal distribution is normalized.

We then consider the remaining terms of f^{p}_{t+1} and define $\left\|\underline{\gamma}^{\mathrm{j}}_{i,:}\right\|_1 = \sum_{k=1}^{n} \gamma^{\mathrm{j}}_{i,k}$ and pull the factor out of the second sum, yielding

$$f^{\mathrm{p}}_{t+1}\left(\underline{x}^{\tau}_{t+1}, \underline{x}^{\omega}_{t+1}\right) = \sum_{i=1}^{n} \mathbb{1}\left\{\underline{x}^{\omega}_{t+1} \in A_i\right\} \frac{\pi^2}{n}$$

$$\sum_{j=1}^{n} \gamma^{\mathrm{j}}_{i,j} f_{\mathrm{N}}\left(\underline{x}^{\tau}_{t+1}; \mathbf{F}_{i,j}\underline{\mu}^{\mathrm{e}}_{j}, +\underline{\hat{u}}^{\tau}_{i,j}, \mathbf{C}^{\underline{\boldsymbol{w}}}_{i,j} + \mathbf{F}_{i,j}\mathbf{C}^{\mathrm{e}}_{j}\mathbf{F}^{\top}_{i,j}\right)$$

$$= \sum_{i=1}^{n} \mathbb{1}\left\{\underline{x}^{\omega}_{t+1} \in A_i\right\} \frac{\pi^2}{n} \left\|\underline{\gamma}^{\mathrm{j}}_{i,:}\right\|_1 \tag{9}$$

$$\sum_{j=1}^{n} \underbrace{\frac{\gamma^{\mathrm{j}}_{i,j}}{\left\|\underline{\gamma}^{\mathrm{j}}_{i,:}\right\|_1}}_{w_{i,j}} f_{\mathrm{N}}\left(\underline{x}^{\tau}_{t+1}; \underbrace{\mathbf{F}_{i,j}\underline{\mu}^{\mathrm{e}}_{j} + \underline{\hat{u}}^{\tau}_{i,j}}_{\underline{v}_{i,j}}, \underbrace{\mathbf{C}^{\underline{\boldsymbol{w}}}_{i,j} + \mathbf{F}_{i,j}\mathbf{C}^{\mathrm{e}}_{j}\mathbf{F}^{\top}_{i,j}}_{\mathbf{U}_{i,j}}\right) . \tag{10}$$

The expression in (9)–(10) is already close to the desired form (3). Each grid value γ^{p}_{i} is $\frac{\pi^2}{n}\left\|\underline{\gamma}^{\mathrm{j}}_{i,:}\right\|_1$. Instead of a single Gaussian, we have a Gaussian mixture with weights $w_{i,j}$, mean vectors $\underline{v}_{i,j}$, and covariance matrices $\mathbf{U}_{i,j}$ as the conditional density in (10).

To get to the desired form (3) that fulfills all assumptions, we perform Gaussian mixture reduction. There are several approaches to Gaussian mixture reduction [19]. We choose moment matching as a highly efficient way. The mean vector $\underline{\mu}^{\mathrm{p}}_{i}$ for the Gaussian for the area i is the weighted combination of the original mean vectors, i.e.,

$$\underline{\mu}^{\mathrm{p}}_{i} = \sum_{j=1}^{n} w_{i,j}\underline{v}_{i,j} .$$

Each predicted covariance depends on both the covariances of the mixture components and the differences in the mean vectors. For this, the sample covariance is added to the weighted combination of the covariances, leading to

$$\mathbf{C}^{\mathrm{p}}_{i} = \left(\sum_{j=1}^{n} w_{i,j}\mathbf{U}_{i,j}\right) + \sum_{j=1}^{n}\left(\underline{v}_{i,j} - \underline{\mu}^{\mathrm{p}}_{i}\right)\left(\underline{v}_{i,j} - \underline{\mu}^{\mathrm{p}}_{i}\right)^{\top}$$

for the predicted covariance for the area i.

IV. Evaluation

In this chapter, we compare the S3F with a PF that we adapted to the SE(3) domain. We consider different number of parameters, i.e., different numbers of areas or particles. Both filters are part of the latest version of libDirectional [20].

A. Scenario Description

We simulate the motion of an object for 10 time steps. The random vector for the initial position $\underline{\boldsymbol{x}}^{\tau}_{1}$ is distributed according to a normal distribution $\mathcal{N}(\underline{0}, \mathbf{I})$, with $\underline{0}$ and $\mathbf{I}$ being the 3-D zero vector and identity matrix, respectively. For the orientation, we constrain a Watson distribution [21, Sec. 9.4.2], which is a antipodally symmetric distribution on the sphere, to the upper hemisphere. This hyperhemispherical Watson distribution is only defined on the upper hemisphere and has twice the density of a regular Watson distribution. The initial prior $\underline{\boldsymbol{x}}^{\omega}_{1}$ is distributed according to a hyperhemispherical Watson distribution $\mathcal{HWD}$ with mean value $\mu = \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^{\top}$ and concentration parameter 1.

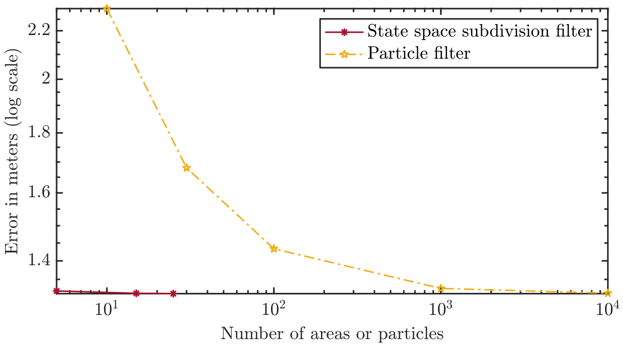

Fig. 1. Errors over number of areas or particles.

The heading angle in time step $t+1$ is distributed according to

$$\underline{\boldsymbol{x}}^{\omega}_{t+1} \sim \mathcal{HWD}(\underline{\boldsymbol{x}}^{\omega}_{t}, 1) .$$

The object moves one unit per time step in the direction it is facing. Moreover, some additive Gaussian noise $\underline{\boldsymbol{w}}^{\tau}_{t} \sim \mathcal{N}(\underline{0}, \mathbf{I})$ is added. Denoting the rotation matrix that corresponds to the quaternion $\underline{q}$ by $\mathbf{R}(\underline{q})$, the motion model for the translatory component is

$$\underline{\boldsymbol{x}}^{\tau}_{t+1} = \underline{\boldsymbol{x}}^{\tau}_{t} + \mathbf{R}(\underline{\boldsymbol{x}}^{\omega}_{t}) + \underline{\boldsymbol{w}}^{\tau}_{t} .$$

It can easily be seen that the next position depends on the current orientation, and thus, the estimation problem cannot be split up into two separate parts for the position and orientation.

As measurements, only position measurements are obtained. The measurements are perturbed by additive white Gaussian noise. The full measurement model is $\underline{\boldsymbol{z}}_{t} = \underline{\boldsymbol{x}}^{\tau}_{t} + \underline{\boldsymbol{v}}_{t}$ with $\underline{\boldsymbol{v}}_{t} \sim \mathcal{N}(\underline{0}, \mathbf{I})$ for all time steps.

B. Evaluation Metrics

4000 runs were performed for all filters. All evaluation metrics are condensed into a single number by determining the average.

One criterion we consider for the two filters is the run time. The run times are given as the average run time per time step, i.e., we divide the total run time of the filter by 10. The run times were measured on a server running Red Hat Enterprise Linux 8.2. Two cores of an Intel Xeon Gold 6230 and 3 GB of RAM were allocated to the job.

For the errors, we consider the Euclidean distance between the estimate provided by the filter (after the update step) and the actual position at the last time step. Because we consider the estimate at the 10th time step and the position and orientation are dependent, the quality of the orientation estimates also plays a role.

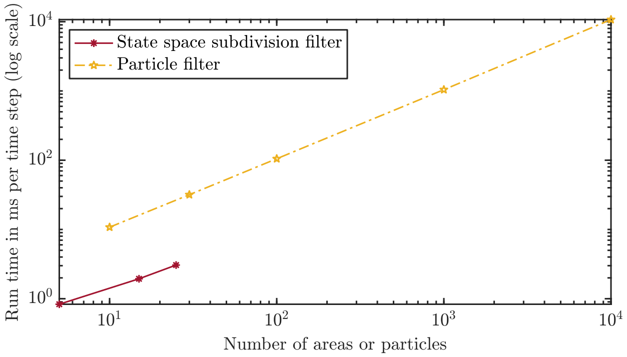

Fig. 2. Run time over number of areas or particles.

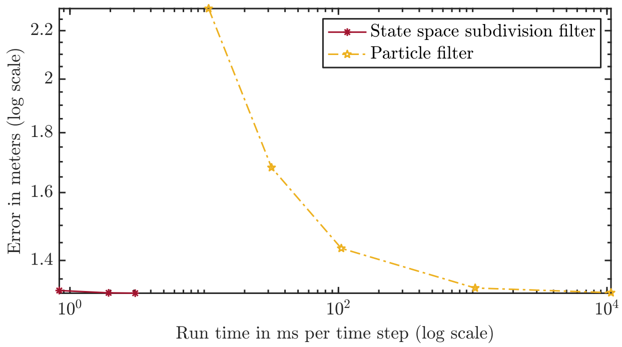

Fig. 3. Error over run time.

C. Evaluation Results

In Fig. 1, the average errors are shown for the two filters and different numbers of parameters. Using only 5 areas, the S3F already yields highly accurate results. The PF requires 10000 particles to achieve an accuracy that is comparable to (but still worse than) that of the S3F using 15 areas. Using 25 areas only leads to an improvement of less than 10^{-3} compared with the S3F using 15 areas. That the S3F does not improve significantly further may indicate that the filter configuration is already close to the accuracy an optimal Bayes filter would achieve.

The run times are shown in Fig. 2. As expected, the run time of the PF scales linearly in the number of particles. In our implementation, the S3F is still faster than the PF for the considered number of parameters, which shows that none of the operations in the S3F is forbiddingly expensive. In the configurations considered, the S3F appears to scale almost linearly in the number of parameters. However, due to its quadratic complexity, one can expect the S3F to have higher run times for higher numbers of parameters.

Since the limiting factor in most cases is run time and not RAM, we also compare the filters on a run time basis in Fig. 3. A configuration that is to the lower left of another led to more accurate results at a lower run time. The configurations of the S3F with 15 and 25 areas are to the lower left of all considered configurations for the PF. Being able to produce more accurate results at lower run times, the S3F is clearly superior to the PF in this scenario.

V. Conclusion and Outlook

In this paper, we presented the S3F for $\mathrm{SE}(3)$. The key idea of the S3F is to split the joint density for the position and orientation up into a marginal part for the orientation and a part for the position that is conditioned on the orientation. While only the unit circle had to be discretized for the $\mathrm{SE}(2)$ variant of the S3F, the S3F for $\mathrm{SE}(3)$ relies on a discrete filter for hyperhemispheres. Each update step is in $O(n)$, and $O(n^2)$ operations are required for the prediction step. Thus, its asymptotic run time behavior is equivalent for that of the original discrete filter for hyperhemispheres.

In our evaluation, the S3F outperformed a PF even for a high number of particles. The S3F provided highly accurate results at low run times and is thus a good option for estimation problems on $\mathrm{SE}(3)$. One limitation is the assumption that the conditional densities are unimodal. However, one could alleviate this issue by allowing for mixtures, which may be the subject of future work. Another limitation is assumption (B1). This assumption does not allow for a direct influence of the current position on the next orientation. If such a dependency is required, one can consider conditioning on the mean of the Gaussian in the respective area or the mean of the marginalized density for the position considering all areas (which is a Gaussian mixture).

As part of future work, one may consider integrating linear and angular velocities into the S3F. While one only has to use higher-dimensional Gaussians for the linear velocities, integrating angular velocities can be more tricky. One way to describe angular velocities is to use a vector in $\mathbb{R}^3$. However, one cannot directly integrate the angular velocity as a quantity in $\mathbb{R}^3$ into the conditional density because assumption (B1) would not allow for considering the angular velocity for the next orientation. One possible solution is to limit the angular velocity, which would lead to a bounded domain. Alternatively, one could consider turn rates greater than one revolution per time step equivalent to a lower turn rate that results in the same orientation in the next time step. For example, if an object turns by $2\pi + 0.1$ per time step around an axis of rotation, the orientation in the next time step will be the same as if the object turned by 0.1 around the same axis. Using such considerations, one can encode turn rates using elements of a manifold of bounded size. Then, one could derive a grid-based filter for the Cartesian product of the manifold for the turn rate and $\mathbb{H}^3$ to simultaneously describe the orientation and turn rate in the marginal density, allowing for the implementation of a good model while preserving assumption (B1). Another area for future work would be to apply the S3F to simultaneous localization and mapping [22] by integrating marker positions into the state.

Acknowledgments

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)*.

References

[1] H. A. Hashim, L. J. Brown, and K. McIsaac, "Nonlinear Stochastic Position and Attitude Filter on the Special Euclidean Group 3," *Journal of the Franklin Institute*, vol. 356, no. 7, pp. 4144–4173, May 2019.

[2] B. Busam, T. Birdal, and N. Navab, “Camera Pose Filtering with Local Regression Geodesics on the Riemannian Manifold of Dual Quaternions,” in *International Conference on Computer Vision 2017 (ICCV 2017)*, Apr. 2017.

[3] S. J. Julier and J. K. Uhlmann, “Unscented Filtering and Nonlinear Estimation,” *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[4] V. Madyastha, V. Ravindra, S. Mallikarjunan, and A. Goyal, “Extended Kalman Filter vs. Error State Kalman Filter for Aircraft Attitude Estimation,” in *AIAA Guidance, Navigation, and Control Conference*, 2011, p. 6615.

[5] J. L. Crassidis and F. L. Markley, “Unscented Filtering for Spacecraft Attitude Estimation,” *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 4, pp. 536–542, 2003.

[6] R. A. Srivatsan, G. T. Rosen, D. F. N. Mohamed, and H. Choset, “Estimating SE(3) Elements Using a Dual Quaternion Based Linear Kalman Filter,” in *Robotics: Science and Systems*, 2016.

[7] H. A. Hashim and F. L. Lewis, “Nonlinear Stochastic Estimators on the Special Euclidean Group SE(3) Using Uncertain IMU and Vision Measurements,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 12, pp. 7587–7600, Dec. 2021.

[8] A. Barrau, “Nonlinear State Error Based Extended Kalman Filters with Applications to Navigation,” Ph.D. dissertation, Mines Paristech, 2015.

[9] A. Barrau and S. Bonnabel, “The Invariant Extended Kalman Filter as a Stable Observer,” *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2017.

[10] S. Bonnabel, “Left-Invariant Extended Kalman Filter and Attitude Estimation,” in *46th IEEE Conference on Decision and Control (CDC 2007)*, 2007, pp. 1027–1032.

[11] S. Bonnable, P. Martin, and E. Salaün, “Invariant Extended Kalman Filter: Theory and Application to a Velocity-Aided Attitude Estimation Problem,” in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC 2009) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 1297–1304.

[12] J.-P. Condomines, C. Seren, and G. Hattenberger, “Nonlinear State Estimation Using an Invariant Unscented Kalman Filter,” in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013.

[13] M. Brossard, A. Barrau, and S. Bonnabel, “A Code for Unscented Kalman Filtering on Manifolds (UKF-M),” in *2020 IEEE International Conference on Robotics and Automation (ICRA 2020)*, May 2020, pp. 5701–5708, iSSN: 2577-087X.

[14] 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.

[15] F. Pfaff, K. Li, and U. D. Hanebeck, “The State Space Subdivision Filter for Estimation on SE(2),” *Sensors*, Sep. 2021.

[16] ——, “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.

[17] ——, “A Hyperhemispherical Grid Filter for Orientation Estimation,” in *Proceedings of the 23rd International Conference on Information Fusion (Fusion 2020)*, Virtual, Jul. 2020.

[18] P. Leopardi, “A Partition of the Unit Sphere into Regions of Equal Area and Small Diameter,” *Electronic Transactions on Numerical Analysis*, vol. 25, no. 12, pp. 309–327, 2006.

[19] D. F. Crouse, P. Willett, K. Pattipati, and L. Svensson, “A Look at Gaussian Mixture Reduction Algorithms,” in *14th International Conference on Information Fusion (FUSION 2011)*. IEEE, 2011.

[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] K. V. Mardia and P. E. Jupp, *Directional Statistics*. John Wiley & Sons, 2000.

[22] T. Bailey and H. Durrant-Whyte, “Simultaneous Localization and Mapping (SLAM): Part II,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.