

Research Article

A New Sparse Gauss-Hermite Cubature Rule Based on Relative-Weight-Ratios for Bearing-Ranging Target Tracking

Lijun Peng, Xiaojun Duan, and Jubo Zhu

College of Science, National University of Defense Technology, Changsha, Hunan 410073, China

Correspondence should be addressed to Lijun Peng; lijunpengzhu@163.com

Received 4 January 2017; Revised 15 June 2017; Accepted 24 July 2017; Published 11 September 2017

Academic Editor: Ming-Cong Deng

Copyright © 2017 Lijun Peng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new sparse Gauss-Hermite cubature rule is designed to avoid dimension explosion caused by the traditional full tensor-product based Gauss-Hermite cubature rule. Although Smolyak's quadrature rule can successfully generate sparse cubature points for high dimensional integral, it has a potential drawback that some cubature points generated by Smolyak's rule have negative weights, which may result in instability for the computation. A relative-weight-ratio criterion based sparse Gauss-Hermite rule is presented in this paper, in which cubature points are kept symmetric in the input space and corresponding weights are guaranteed to be positive. The generation of the new sparse cubature points set is simple and meaningful for practice. The difference between our new sparse Gauss-Hermite cubature rule and other cubature rules is analysed. Simulation results show that, compared with Kalman filter with those types of full tensor-product based Gauss-Hermite rules, our new sparse Gauss-Hermite cubature rule based Kalman filter can lead to a substantially reduced number of cubature points, more stable computation capability, and maintaining the accuracy of integration at the same time.

1. Introduction

Bayesian recursive estimation is commonly used in target tracking, positioning, and signal processing [1]. Generally, a Bayesian recursive estimation algorithm requires a state model and a measurement model. The a posteriori density function can describe the behaviour of the estimated state [2]. Since a closed form solution to the Bayesian recursive estimation is available only for a few special cases [3], such as the linear Gaussian system (which leads to the classical standard Kalman filter), a suboptimal solution is a preferable choice in the general case [4, 5]. Ito and Xiong [4] suggest that a local Gaussian filter can be used to approximate the general Bayesian recursive estimation suboptimally. The core problem of local Gaussian filters is in fact a high dimensional Gaussian weighted integration, which has been studied in numerical analysis; see [6–9] and the references therein. Lots of researches concentrate on the numerical approximation methods to solve the Gaussian weighted integral problem, and different approaches result in different local Gaussian filters, such as the cubature Kalman filter [10], the quadrature Kalman filter [11], and related variants [12–14].

Gauss-Hermite filter, introduced by [2, 4], makes use of the Gauss-Hermite quadrature rule and has the highest accuracy among all the above filters. However, it suffers from the curse of dimensionality since the number of cubature points increases exponentially with the dimension of the state.

To avoid the curse of dimensionality, several sparse rules are developed. Smolyak rule [15] is one of the useful tools to generate a small number of cubature points for high dimensional integral. The computational cost does not increase exponentially by using this sparse grid method. Reference [16] compares the approximation accuracies of various sparse grid methods, including trapezoidal rule, Crenshaw Curtis rule, and Gauss Patterson rule. Reference [17] combines Smolyak's rule and Gaussian Quadrature rule for high dimensional likelihood function in economic models. Jia et al. [18, 19] make a tracking comparison among Smolyak's rule based Gauss-Hermite filter of different levels and the traditional cubature filters in the context of determination of the spacecraft attitude and the lower-earth orbit satellite orbit. Simulations prove the effectiveness of Smolyak's rule based Gauss-Hermite filter. Reference [20] proposes the multiple sparse grid Gauss-Hermite filter based on sparse

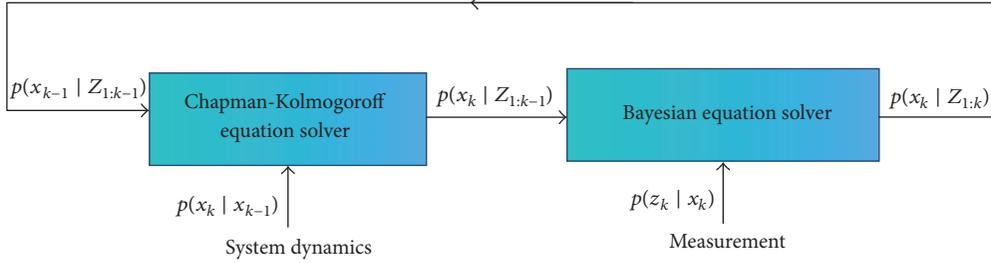


FIGURE 1: One cycle of Bayesian recursive estimation algorithm [18].

grid Gauss-Hermite filter and state-space partitioning. It is claimed that the computational burden is further reduced with respect to the Gauss-Hermite filter and the sparse grid Gauss-Hermite filter.

However, as Heiss and Winschel [17] mentioned, there exists a potential drawback that some of the cubature points have negative weights, which may result in instability for computation. So we come up with the idea to avoid this potential drawback. In this paper, a classification of full tensor-product Gauss-Hermite cubature points is obtained by using equivalence classification by position permutation and signum function. There exist plenty of cubature points with low weights, which can be deleted from the cubature scheme directly considering a relative-weight-ratio threshold. Comparing with the Smolyak-based rule, our construction of the new Gauss-Hermite rule is simple and practically meaningful. The corresponding weights of the cubature points are all positive; meanwhile, the full symmetry property still remains. There also exist some interesting relationships among the Gauss-Hermite filter, the 3rd embedding cubature filter, the sparse Gauss-Hermite filter, and the Unscented Kalman filter.

The remainder of this paper is organized as follows: In Section 2, a brief review of the nonlinear system and its Bayesian recursive estimation framework is presented. In Section 3, a cycle of general local Gaussian filter is presented, which offers six kinds of Gaussian weighted integrals. This leads to various specific local Gaussian filters by various choice of the cubature points set and the corresponding weights. In Section 4, a full tensor-product Gauss-Hermite integral cubature rule is presented. By introducing a simple and elegant operator, the full tensor-product based cubature points can be divided into different categories. A small set of positive weighted cubature points is generated by a threshold of relative-weight-ratio. Relationships between the new sparse Gauss-Hermite filter and the traditional cubature filters are analysed as well. In Section 5, a typical Bearing-Ranging tracking problem with a general 2D manoeuvring target motion is demonstrated to test the performance of our new sparse Gauss-Hermite filter. Some conclusions are given in Section 6.

2. Bayesian Recursive Estimation Algorithm

Consider the following discrete nonlinear system:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1}, \quad (1)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (2)$$

where (1) and (2) are the motion model and the measurement model of the system, respectively; $\mathbf{x}_k \in \mathbb{R}^n$ is the state of the system and $\mathbf{z}_k \in \mathbb{R}^m$ is the measurement; $\mathbf{w}_k \in N(0, \mathbf{Q}_k)$ is the process noise and $\mathbf{v}_k \in N(0, \mathbf{R}_k)$ is the measurement noise. \mathbf{x}_k is independent with \mathbf{w}_k and \mathbf{v}_k . By the discrete time Chapman-Kolmogoroff equation,

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{Z}_{1:k-1}) \\ = \int_{\mathbb{R}^n} p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{Z}_{1:k-1}) d\mathbf{x}_{k-1}, \end{aligned} \quad (3)$$

where $\mathbf{Z}_{1:k-1} = \{\mathbf{z}_1, \dots, \mathbf{z}_{k-1}\}$. By the Bayesian formula,

$$p(\mathbf{x}_k | \mathbf{Z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Z}_{1:k-1})}{\int_{\mathbb{R}^n} p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{Z}_{1:k-1}) d\mathbf{x}_k}, \quad (4)$$

where $\mathbf{Z}_{1:k} = \{\mathbf{z}_1, \dots, \mathbf{z}_k\}$. Equations (3) and (4) are the time update formula and the measurement update formula. Figure 1 shows the recursive Bayesian estimation algorithm. When $p(\mathbf{x}_k | \mathbf{Z}_{1:k})$ is obtained, we can estimate the state $\hat{\mathbf{x}}_k$ and its covariance matrix \mathbf{P}_k by minimizing the mean square error. The results are presented as follows:

$$\begin{aligned} \hat{\mathbf{x}}_k &= E(\mathbf{x}_k | \mathbf{Z}_{1:k}) = \int_{\mathbb{R}^n} \mathbf{x}_k p(\mathbf{x}_k | \mathbf{Z}_{1:k}) d\mathbf{x}_k, \\ \mathbf{P}_k &= E((\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T | \mathbf{Z}_{1:k}) \\ &= \int_{\mathbb{R}^n} (\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T p(\mathbf{x}_k | \mathbf{Z}_{1:k}) d\mathbf{x}_k. \end{aligned} \quad (5)$$

However, when $p(\mathbf{x}_k | \mathbf{Z}_{1:k})$ becomes an arbitrary probability density distribution, it is difficult to calculate the high dimensional integrations in (5) directly. A suboptimal way is to place normal distributions on $p(\mathbf{x}_{k-1} | \mathbf{Z}_{1:k-1})$, $p(\mathbf{x}_k | \mathbf{Z}_{1:k-1})$, and $p(\mathbf{x}_k | \mathbf{Z}_{1:k})$; thus the first-order moment and second-order moment can be used to describe the posterior probability. In other words, at time epoch $k-1$, the posterior density is

$$p(\mathbf{x}_{k-1} | \mathbf{Z}_{1:k-1}) = N(\mathbf{x}_{k-1} : \hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}). \quad (6)$$

At time epoch k , the predicted probability density of the state is

$$p(\mathbf{x}_k | \mathbf{Z}_{1:k-1}) = N(\mathbf{x}_k : \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}). \quad (7)$$

Correspondingly, the posterior density of the state is

$$p(\mathbf{x}_k | \mathbf{Z}_{1:k}) = N(\mathbf{x}_k : \hat{\mathbf{x}}_k, \mathbf{P}_k). \quad (8)$$

3. Local Gaussian Filter

Section 2 points out that, under the Gaussian assumption, the first-order moment and second-order moment can be used to describe the property of the state. Hence, a suboptimal local Gaussian filter algorithm can replace the Bayesian recursive estimation (3) and (4). Details of general Gaussian filter can be found in [4].

(1) Time update.

(i) Prediction of state $\hat{\mathbf{x}}_{k|k-1}$:

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= E[f(\mathbf{x}_{k-1}) + \mathbf{w}_k | \mathbf{Z}_{1:k-1}] \\ &= \int_{\mathbb{R}^n} f(\mathbf{x}_{k-1}) N(\mathbf{x}_{k-1} : \hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}) d\mathbf{x}_{k-1}.\end{aligned}\quad (9)$$

(ii) Prediction of covariance matrix $\mathbf{P}_{k|k-1}$ given state $\hat{\mathbf{x}}_{k|k-1}$:

$$\begin{aligned}\mathbf{P}_{k|k-1} &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) \cdot (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})^T | \mathbf{Z}_{1:k-1}] \\ &= E[(f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} - \hat{\mathbf{x}}_{k|k-1}) \\ &\quad \cdot (f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} - \hat{\mathbf{x}}_{k|k-1})^T | \mathbf{Z}_{1:k-1}] = \mathbf{Q}_k \\ &\quad + \int_{\mathbb{R}^n} (f(\mathbf{x}_{k-1}) - \hat{\mathbf{x}}_{k|k-1}) \cdot (f(\mathbf{x}_{k-1}) - \hat{\mathbf{x}}_{k|k-1})^T \\ &\quad \cdot N(\mathbf{x}_{k-1} : \hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}) d\mathbf{x}_{k-1}.\end{aligned}\quad (10)$$

(2) Measurement update.

(i) Prediction of measurement $\hat{\mathbf{z}}_{k|k-1}$:

$$\begin{aligned}\hat{\mathbf{z}}_{k|k-1} &= E[h(\mathbf{x}_k) + \mathbf{v}_k | \mathbf{Z}_{1:k-1}] \\ &= \int_{\mathbb{R}^m} h(\mathbf{x}_k) N(\mathbf{x}_k : \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) d\mathbf{x}_k.\end{aligned}\quad (11)$$

(ii) Covariance matrix $\mathbf{P}_{k|k-1}^{zz}$ given $\hat{\mathbf{z}}_{k|k-1}$:

$$\begin{aligned}\mathbf{P}_{k|k-1}^{zz} &= E[(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T | \mathbf{Z}_{1:k-1}] \\ &= E[(\mathbf{v}_k + h(\mathbf{x}_k) - \hat{\mathbf{z}}_{k|k-1}) \\ &\quad \cdot (\mathbf{v}_k + h(\mathbf{x}_k) - \hat{\mathbf{z}}_{k|k-1})^T | \mathbf{Z}_{1:k-1}] = \mathbf{R}_k \\ &\quad + \int_{\mathbb{R}^n} (h(\mathbf{x}_k) - \hat{\mathbf{z}}_{k|k-1})(h(\mathbf{x}_k) - \hat{\mathbf{z}}_{k|k-1})^T \\ &\quad \cdot N(\mathbf{x}_k : \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) d\mathbf{x}_k.\end{aligned}\quad (12)$$

(iii) Covariance matrix $\mathbf{P}_{k|k-1}^{xz}$ between state $\hat{\mathbf{x}}_{k|k-1}$ and measurement $\hat{\mathbf{z}}_{k|k-1}$:

$$\begin{aligned}\mathbf{P}_{k|k-1}^{xz} &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1})^T | \mathbf{Z}_{1:k-1}] \\ &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(h(\mathbf{x}_k) + \mathbf{v}_k - \hat{\mathbf{z}}_{k|k-1})^T | \mathbf{Z}_{1:k-1}] \\ &= \int_{\mathbb{R}^n} (\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(h(\mathbf{x}_k) - \hat{\mathbf{z}}_{k|k-1})^T \\ &\quad \cdot N(\mathbf{x}_k : \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) d\mathbf{x}_k.\end{aligned}\quad (13)$$

(iv) Gain of the filter:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}^{xz} (\mathbf{P}_{k|k-1}^{zz})^{-1}.\quad (14)$$

(v) State update $\hat{\mathbf{x}}_k$ and its covariance matrix \mathbf{P}_k :

$$\begin{aligned}\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}), \\ \mathbf{P}_k &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{k|k-1}^{zz} \mathbf{K}_k^T.\end{aligned}\quad (15)$$

We can see that (9), (10), (14), and (15) are similar to those corresponding formulae of standard Kalman filter. The only difference between local Gaussian filter and standard Kalman filter is the calculation of $\hat{\mathbf{x}}_{k|k-1}$, $\hat{\mathbf{z}}_{k|k-1}$, $\mathbf{P}_{k|k-1}$, $\mathbf{P}_{k|k-1}^{zz}$, and $\mathbf{P}_{k|k-1}^{xz}$. For standard Kalman filter, there exist specific formulae, while for the local Gaussian filter, numerical approximation methods need to calculate the higher dimensional integrals. A general high dimensional integral with Gaussian type weighting function can be expressed as follows:

$$\begin{aligned}I &= \int_{\mathbb{R}^n} F(\mathbf{t}) \frac{1}{((2\pi)^n \det \Sigma)^{1/2}} \\ &\quad \times \exp\left[\frac{1}{2}(\mathbf{t} - \bar{\mathbf{x}})^T \Sigma^{-1}(\mathbf{t} - \bar{\mathbf{x}})\right] d\mathbf{t}.\end{aligned}\quad (16)$$

Suppose that $\Sigma = \mathbf{S}^T \mathbf{S}$, where \mathbf{S} is a lower triangular matrix. By a linear transform $\mathbf{t} = \mathbf{S}^T \mathbf{s} + \bar{\mathbf{x}}$, we can get that

$$\begin{aligned}I &= \int_{\mathbb{R}^n} F(\mathbf{S}^T \mathbf{s} + \bar{\mathbf{x}}) \frac{1}{(2\pi)^{n/2}} e^{-(1/2)|\mathbf{s}|^2} d\mathbf{s} \\ &\approx \sum_i^N w_i F(\mathbf{S}^T \xi_i + \bar{\mathbf{x}}).\end{aligned}\quad (17)$$

Various methods can approximate (17), such as the spherical radial integral rule [7, 21], Gauss-Hermite rule [2], central differential rule [4], Stochastic Integral Rule [22], Gaussian localized cumulative distribution rule [23], and Smolyak rule based sparse grid rule [18]. All of them use specific ways to generate $\{\xi_i\}_{i=1}^N$ and their corresponding weights $\{w_i\}_{i=1}^N$, where N is the number of cubature points set.

Remark 1. Generally, the choice of $\{\xi_i, w_i\}_{i=1}^N$ has no relationship with the estimation of state $\hat{\mathbf{x}}_k$ and its covariance matrix \mathbf{P}_k , but uniquely determined by the specific cubature integral rule. In fact, almost all cubature rules based $\{\xi_i, w_i\}_{i=1}^N$ can be computed and stored offline.

4. Sparse Gauss-Hermite Integral Rules

4.1. Symmetry Set Based on Full Position Permutation

Definition 2. In order to utilize the symmetry property of the cubature points, $[\bullet]_n$ is introduced to achieve a more concise denotation of the cubature points set. Denote \bullet as a nonnegative seed; we have its corresponding seed vector $\nu(\bullet)$ as a n -dimensional vector $(\bullet, \mathbf{0})^T$, where n is the state

TABLE 1: Cubature points and corresponding weights for one dimension ($m = 1, 2, 3, 4, 5$).

m	$\{\xi\}$	$\{w\}$
1	$\{0\}$	$\{1\}$
2	$\{-1, 1\}$	$\left\{\frac{1}{2}, \frac{1}{2}\right\}$
3	$\{-\sqrt{3}, 0, +\sqrt{3}\}$	$\left\{\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right\}$
4	$\{-2.334, -0.742, 0.742, 2.334\}$	$\{0.046, 0.454, 0.454, 0.046\}$
5	$\{-2.857, -1.356, 0, 1.356, 2.857\}$	$\{0.0113, 0.2221, 0.5333, 0.2221, 0.0113\}$

dimension. Then we have $[\bullet]_n = \{\text{FP}(v(\bullet)), -\text{FP}(v(\bullet))\}$, where $\text{FP}(v(\bullet))$ means that the set of elements consists of the full permutation vectors of the seed vector and $-\text{FP}(v(\bullet))$ is generated by randomly setting one or several nonzero values to their opposites in $\text{FP}(v(\bullet))$.

Let the state dimension be n and take $[1, 1]_n$ as an example, where $\{1, 1\}$ is the seed. Since two values are given, then the n -dimensional seed vector is $(1, 1, 0, \dots, 0)^T$. It has total $\binom{n}{2}$ full permutation, so $[1, 1]_n$ has $\binom{n}{2} + \binom{2}{1} \binom{n}{2} + \binom{2}{2} \binom{n}{2}$ elements. For simplicity, suppose $n = 3$; then $[1]_3$ has $\binom{3}{1} + \binom{3}{1}$ elements, which is presented as the following set:

$$[1]_3 = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \right\}. \quad (18)$$

Remark 3. With the help of full position permutation, it becomes much easier to describe the cubature points set $\{\xi_i, i = 1, 2, \dots, N\}$. Furthermore, it brings some interesting properties for the cubature points.

4.2. Full Tensor-Product Based Gauss-Hermite Integral Rule. The classical one-dimension Gauss-Hermite rule can be easily obtained according to [24]. Table 1 presents us with cubature points set $\{\xi\}$ and corresponding weights $\{w\}$ from orders 1 to 5. Here the order m also represents the number of cubature points in each dimension. We can see that the points and weights are both symmetric in \mathbb{R} . For details about the construction of Gauss-Hermite quadrature points, one can refer to [25]. Furthermore, [26] provides a better and more stable way to construct cubature points and weights than moment matching method.

For n dimension integral problem, a full tensor-product based Gauss-Hermite rule can be obtained directly by the following formula [2]:

$$\int_{\mathbb{R}^n} f(\mathbf{x}) N(\mathbf{x}, 0, \mathbf{I}_n) d\mathbf{x} \approx \sum_{i_1=1}^m w_{i_1} \cdots \sum_{i_n=1}^m w_{i_n} f(\xi_{i_1}, \dots, \xi_{i_n}) = \sum_{i=1}^{m^n} w_i f(\xi_i). \quad (19)$$

As mentioned above, n is the dimension of the state, m is the number of the cubature points for each dimension, and $1 \leq i \leq m^n$ as presented in (19). Each point of the cubature set is a n -dimensional vector; that is, $\xi_i = [\xi_{i_1}, \dots, \xi_{i_n}]^T$. The weight of point i can be calculated as $w_i = \prod_{p=1}^n w_{i_p}$. For example, let $m = 3$ and $n = 2$; we can easily get the cubature points and the corresponding weights as follows:

$$\begin{aligned} \xi &= \{(0, 0), (0, -\sqrt{3}), (0, \sqrt{3}), (\sqrt{3}, 0), (\sqrt{3}, \sqrt{3}), \\ &\quad (\sqrt{3}, -\sqrt{3}), (-\sqrt{3}, 0), (-\sqrt{3}, \sqrt{3}), (-\sqrt{3}, -\sqrt{3})\}, \quad (20) \\ \mathbf{w} &= \left\{ \frac{4}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{36}, \frac{1}{36}, \frac{1}{9}, \frac{1}{36}, \frac{1}{36} \right\}. \end{aligned}$$

Recall Definition 2; the cubature points in (20) can be categorized as three sets $\{\xi_0, \xi_1, \xi_2\}$; that is,

$$\begin{aligned} \xi_0 &= [0]_2, \\ \xi_1 &= [\sqrt{3}]_2, \\ \xi_2 &= [\sqrt{3}, \sqrt{3}]_2, \end{aligned} \quad (21)$$

and the weights can be summarized as three types as well:

$$\begin{aligned} w_0 &= \frac{4}{9}, \\ w_1 &= \frac{1}{9}, \\ w_2 &= \frac{1}{36}. \end{aligned} \quad (22)$$

The number of each category can be easily computed as

$$\begin{aligned} n_0 &= 1, \\ n_1 &= \binom{2}{1} + \binom{2}{1} = 4, \\ n_2 &= \binom{2}{2} + \binom{2}{1} \binom{2}{2} + \binom{2}{2} \binom{2}{2} = 4. \end{aligned} \quad (23)$$

The total number of cubature points $N = n_0 + n_1 + n_2 = 9 = 3^2$. Equations (21), (22), and (23) show that such a classification is an equivalent notation of the full tensor-product rule. Now suppose that $m = 3$ remains unchanged; we would like

to consider the general case of n dimension. According to Definition 2, there exist $n + 1$ categories. The corresponding $\{\xi_i, i = 0, 1, 2, \dots, n\}$, $\{w_i, i = 0, 1, 2, \dots, n\}$, and the number of each category $\{n_i, i = 0, 1, 2, \dots, n\}$ are shown as follows:

$$\begin{aligned}
\xi_0 &= [0]_n \\
\xi_1 &= [\sqrt{3}]_n \\
\xi_2 &= [\sqrt{3}, \sqrt{3}]_n \\
&\vdots \\
\xi_n &= [\sqrt{3}, \dots, \sqrt{3}]_n \\
w_0 &= \left(\frac{2}{3}\right)^n \\
w_1 &= \left(\frac{2}{3}\right)^{n-1} \frac{1}{6} \\
w_2 &= \left(\frac{2}{3}\right)^{n-2} \left(\frac{1}{6}\right)^2 \\
&\vdots \\
w_n &= \left(\frac{1}{6}\right)^n \\
n_0 &= 1 \\
n_1 &= \binom{n}{1} + \binom{n}{1} = 2n \\
n_2 &= \binom{n}{2} \left(\binom{2}{0} + \binom{2}{1} + \binom{2}{2} \right) = 2n(n-1) \\
n_3 &= \binom{n}{3} \left(\binom{3}{0} + \binom{3}{1} + \binom{3}{2} + \binom{3}{3} \right) \\
&= \frac{4n(n-1)(n-2)}{3} \\
&\vdots \\
n_n &= \binom{n}{n} \left(\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n} \right) = 2^n.
\end{aligned} \tag{24}$$

The total number of cubature points $N = n_0 + n_1 + \dots + n_n = 3^n$. It also shows that such an expression is another form of the full tensor-product based rule. When the dimension of state n increases, the number of cubature points N increases exponentially with n . The curse of dimensionality limits the usage of Gauss-Hermite integral rule.

Now, we are going to give the cubature points set and their corresponding weights with $m = 1, 2, 4, 5$. It can be easily conducted that when $m = 1$, there is only one cubature point; that is, $\xi = (0, 0, \dots, 0)$, and its weight is 1. When $m = 2$, we

can refer to Table 1 to find that there is only one nonnegative value (i.e., 1), so there exists only one category as well:

$$\begin{aligned}
\xi_0 &= [1, 1, \dots, 1]_n; \\
w_0 &= \left(\frac{1}{2}\right)^n, \\
n_0 &= 2^n.
\end{aligned} \tag{25}$$

When $m = 2$, it is interesting to find that the GHKF is similar to the 3rd embedding cubature Kalman filter [27]. We will elaborate their connections in the following Remark.

Remark 4. The main difference between GHKF and the 3rd embedding cubature Kalman filter is the central point $(0, 0, \dots, 0)^T$ and its weight w_0 utilized. The classical standard 3rd cubature rule based on spherical radial rule [10, 28], whose cubature points set is $[\sqrt{n}]$, may exceed the range of state. Both the GHKF and the 3rd embedding cubature rule are stable for high dimensional integral rule. However, since the number of cubature points in 3rd embedding cubature Kalman filter is $2^n + 1$, we still face the threat of dimension explosion when dealing with high dimensional system.

When $m = 4$, there still exists $n + 1$ type of category which is analogous to the case $m = 3$. Here we can simply regard $p_{4,1}$ as 0 of case $m = 3$ only to remember that $p_{4,1}$ is positive. We have $p_{4,1} = 0.7442$ and $p_{4,2} = 1.3556$ according to Table 1. The total number of cubature points becomes 4^n . More specifically, the cubature points set $\{\xi\}$, the corresponding weights $\{w\}$, and the number $\{n_i\}$ for each category are shown as follows:

$$\begin{aligned}
\xi_0 &= [p_{4,1}, p_{4,1}, \dots, p_{4,1}]_n \\
\xi_1 &= [p_{4,1}, \dots, p_{4,1}, p_{4,2}]_n \\
&\vdots \\
\xi_n &= [p_{4,2}, p_{4,2}, \dots, p_{4,2}]_n \\
w_0 &= (w_{4,1})^n \\
w_1 &= (w_{4,1})^{n-1} w_{4,2} \\
w_2 &= (w_{4,1})^{n-2} w_{4,2}^2 \\
&\vdots \\
w_n &= w_{4,2}^n \\
n_0 &= 2^n \\
n_1 &= 2^n \left(\binom{n}{1} + \binom{n}{1} \right) = 2^n \cdot 2 \\
n_2 &= 2^n \cdot 2n(n-1) \\
&\vdots \\
n_4 &= 2^n,
\end{aligned} \tag{26}$$

where $w_{4,1} = 0.4541$, $w_{4,2} = 0.0459$. It can be seen that the number of each cubature category has a factor 2^n , which will result in the dimension explosion as well.

When $N = 5$, the total number of cubature points is $N = 5^n$. It can be classified as follows:

$$\begin{aligned}
 \xi_0 &= [0]_n \\
 \xi_1 &= [p_{5,1}]_n \\
 \xi_2 &= [p_{5,2}]_n \\
 \xi_3 &= [p_{5,1}, p_{5,1}]_n \\
 \xi_4 &= [p_{5,1}, p_{5,2}]_n \\
 \xi_5 &= [p_{5,2}, p_{5,2}]_n \\
 \xi_6 &= [p_{5,1}, p_{5,1}, p_{5,1}]_n \\
 \xi_7 &= [p_{5,1}, p_{5,1}, p_{5,2}]_n \\
 \xi_8 &= [p_{5,1}, p_{5,2}, p_{5,2}]_n \\
 \xi_9 &= [p_{5,2}, p_{5,2}, p_{5,2}]_n \\
 &\vdots
 \end{aligned} \tag{27}$$

The corresponding weights are presented as

$$\begin{aligned}
 w_0 &= w_{5,0}^n \\
 w_1 &= w_{5,0}^{n-1} w_{5,1} \\
 w_2 &= w_{5,0}^{n-1} w_{5,2} \\
 w_3 &= w_{5,0}^{n-2} w_{5,1}^2 \\
 w_4 &= w_{5,0}^{n-2} w_{5,1} w_{5,2} \\
 w_5 &= w_{5,0}^{n-2} w_{5,2}^2 \\
 w_6 &= w_{5,0}^{n-3} w_{5,1}^3 \\
 w_7 &= w_{5,0}^{n-3} w_{5,1}^2 w_{5,2} \\
 w_8 &= w_{5,0}^{n-3} w_{5,1} w_{5,2}^2 \\
 w_9 &= w_{5,0}^{n-3} w_{5,2}^3 \\
 &\vdots
 \end{aligned} \tag{28}$$

where $p_{5,1} = 1.3556$, $p_{5,2} = 2.857$. $w_0 = 0.5333$, $w_{4,1} = 0.2221$, $w_{4,2} = 0.1123$. We can calculate the number of each category to obtain the following equations:

$$\begin{aligned}
 n_0 &= 1 \\
 n_1 &= 2n \\
 n_2 &= 2n
 \end{aligned}$$

$$\begin{aligned}
 n_3 &= 2n(n-1) \\
 n_4 &= 2n(n-1) \\
 n_5 &= 2n(n-1) \\
 n_6 &= \frac{4n(n-1)(n-2)}{3} \\
 n_7 &= \frac{4n(n-1)(n-2)}{3} \\
 n_8 &= \frac{4n(n-1)(n-2)}{3} \\
 n_9 &= \frac{4n(n-1)(n-2)}{3} \\
 &\vdots
 \end{aligned} \tag{29}$$

Here we can see that, for $m = 2, 4$, the number of each category has a factor 2^n , so those cubature points sets are not a good choice for the high dimensional system. However, the number of each category is a polynomial of the dimension of state when $m = 3, 5$. In the face of the dimension explosion, it is rather tempting that we can just choose several categories to approximate the high dimensional integration. We have introduced the relative-weight-ratio as a criterion to achieve this goal. Detailed analysis is presented in the following subsection.

4.3. Sparse Gauss-Hermite Rule Based on Relative-Weight-Ratio. Take $m = 3$ as an example and recall (21) and (22) which present the cubature points set and corresponding weights set, respectively. The relative-weight-ratio between category $\xi_0 = [0]_n$ and category $\xi_k = [\sqrt{3}, \dots, \sqrt{3}]_n$ can be easily obtained:

$$\text{Ratio}_k = \frac{w_0}{w_k} = 4^k \quad k \geq 0. \tag{30}$$

Equation (30) shows that as the subindex k of the cubature points ξ_k set increases, the Ratio_k decreases exponentially. It indicates that when considering a certain degree of accuracy, there is no need to use all Gauss-Hermite cubature points. When a threshold value K is settled, only the categories satisfying $\text{Ratio}_k \leq K$ will be sufficient for the high dimensional integral. For example, set $K = 100$; then we have

$$\begin{aligned}
 \frac{w_0}{w_3} &= 64 < 100, \\
 \frac{w_0}{w_4} &= 128 > 100.
 \end{aligned} \tag{31}$$

According to the criterion, the categories including $[0]_n$, $[\sqrt{3}]_n$, $[\sqrt{3}, \sqrt{3}]_n$, and $[\sqrt{3}, \sqrt{3}, \sqrt{3}]_n$ are enough to approximate the integral. The total number of the cubature points is

$$N = 1 + 2n + 2n(n-1) + \frac{4n(n-1)(n-1)}{3}. \tag{32}$$

The number of cubature points has a polynomial relationship with the dimension of state. If we set threshold value as

$K = 50$, the remaining categories will become $[0]_n$, $[\sqrt{3}]_n$, and $[\sqrt{3}, \sqrt{3}]_n$. The corresponding number of the cubature points is

$$N = 1 + 2n + 2n(n - 1). \quad (33)$$

Remark 5. By setting the ratio threshold K , most low weighted cubature points will be eliminated. Such a strategy can efficiently eliminate the problem of dimension explosion when dealing with high dimensional system. Bear in mind that since lots of low weighted cubature points are eliminated, the sum of remaining points' weights does not equal to 1. We can deal with this problem by renormalizing the remaining weights and meanwhile maintaining the relative-weight-ratio. Unlike the Unscented Transform rules [29] or the higher order spherical radial rule cubature rule [28], our transformation keeps all weights positive.

Remark 6. When $m = 3$, considering $[0]_n$ and $[\sqrt{3}]_n$ as selected cubature points, the total number is $2n + 1$ which equals the number of the UKF with parameter $\kappa = 3 - n$ [29]. The difference lies in the determination of weights of the cubature points. In the context of sparse GHKF, the weight of $[0]_n$ is $w_{0,\text{GHKF}} = 4/2n + 4$. However under the background of UKF, the weight becomes $w_{0,\text{UKF}} = 1 - n/3$. Similarly, we have weights of $[\sqrt{3}]_n$ as $w_{1,\text{GHKF}} = 1/2n + 4$ and $w_{1,\text{UKF}} = 1/6$ for sparse GHKF and UKF, respectively. When dimension $n > 3$, the weight of $[0]_n$ becomes negative in the UKF context, which is a potential unstable factor for integral approximation. The sparse Gauss-Hermite cubature points for $m = 5$ can also be obtained by repeating the above discussed process with a preset relative-weight-ratio. Comparing $[p_{5,1}, p_{5,1}]_n$ with $[p_{5,2}]_n$, the weight of the former category is larger than the latter one.

5. Numerical Investigation

In this section, a Nearly Constant Turning (NCT) manoeuvring target model is settled to verify the tracking performance of our new sparse Gauss-Hermite Kalman filter. NCT model is commonly used in aircraft flying control. We consider the 2-dimensional movement. Suppose the unknown turning rate is ω . Since ω is unknown, the NCT model is a time-varying nonlinear model [10]. The discrete state model can be settled as follows:

$$\begin{aligned} \mathbf{x}_k &= \begin{bmatrix} 1 & \frac{\sin(\omega_{k-1}\Delta t)}{\omega_{k-1}} & 0 & \frac{\cos(\omega_{k-1}\Delta t) - 1}{\omega_{k-1}} & 0 \\ 0 & \frac{\cos(\omega_{k-1}\Delta t)}{\omega_{k-1}} & 0 & -\sin(\omega_{k-1}\Delta t) & 0 \\ 0 & \frac{1 - \cos(\omega_{k-1}\Delta t)}{\omega_{k-1}} & 1 & \frac{\sin(\omega_{k-1}\Delta t)}{\omega_{k-1}} & 0 \\ 0 & \sin(\omega_{k-1}\Delta t) & 0 & \cos(\omega_{k-1}\Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} \quad (34) \\ &+ \mathbf{w}_k, \end{aligned}$$

TABLE 2: Parameter setting for the experiment.

ΔT	1 s
w_0	-3 deg/s
q_x, q_y	$0.1 \text{ m}^2/\text{s}^3$
q_w	$1.75 \times 10^{-4} \text{ s}^{-3}$
σ_r	10 m
σ_θ	$\sqrt{10}$ mrad

where $\mathbf{x}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k \ \omega_k]^T$. x_k and \dot{x}_k represent the position and velocity of axis x , respectively. y_k and \dot{y}_k represent the position and velocity of axis y , respectively. ΔT is the sampling time interval. ω_k is the turning rate at time epoch k . The corresponding process noise $\mathbf{w}_k \approx N(0, \mathbf{Q})$, where \mathbf{Q} can be rewritten as follows:

$$\mathbf{Q} = \begin{bmatrix} q_x \frac{\Delta T^3}{3} & q_x \frac{\Delta T^2}{2} & 0 & 0 & 0 \\ q_x \frac{\Delta T^2}{2} & q_x \Delta T & 0 & 0 & 0 \\ 0 & 0 & q_y \frac{\Delta T^3}{3} & q_y \frac{\Delta T^2}{2} & 0 \\ 0 & 0 & q_y \frac{\Delta T^2}{2} & q_y \Delta T & 0 \\ 0 & 0 & 0 & 0 & q_w \Delta T \end{bmatrix}. \quad (35)$$

q_x and q_y represent the Power Spectrum Density for positions x and y . q_w is the Power Spectrum Density of the turning rate. Suppose that the sensor platform is settled at the origin points. Bearing-Ranging measurement is used for the target tracking. The measurement equation can be expressed as follows:

$$\mathbf{z}_k = \begin{bmatrix} \sqrt{x_{1,k}^2 + x_{3,k}^2} \\ \tan^{-1} \left(\frac{x_{1,k}}{x_{3,k}} \right) \end{bmatrix} + \mathbf{v}_k, \quad (36)$$

where \mathbf{v}_k is the measurement noise, which satisfies the Gaussian distribution with zero mean. The corresponding covariance matrix \mathbf{R} can be expressed as follows:

$$\mathbf{R} = \text{diag}([\sigma_r^2 \ \sigma_\theta^2]). \quad (37)$$

The corresponding parameters are settled by Table 2 [10].

We set the initial state of the target \mathbf{x}_0 as

$$\begin{aligned} \mathbf{x}_0 &= [1000 \text{ m} \ 300 \text{ ms}^{-1} \ 1000 \text{ m} \ 0 \text{ ms}^{-1} \ -3 \text{ degs}^{-1}]. \end{aligned} \quad (38)$$

Then the corresponding covariance matrix \mathbf{P}_0 of the initial state is

$$\begin{aligned} \mathbf{P}_0 &= \text{diag}[100 \text{ m}^2 \ 10 \text{ m}^2 \text{s}^{-2} \ 100 \text{ m}^2 \ 10 \text{ m}^2 \text{s}^{-2} \ 100 \text{ mrad}^2 \text{s}^{-2}]. \end{aligned} \quad (39)$$

The estimate of initial state of the filter $\hat{\mathbf{x}}_0$ is randomly generated by the distribution $N(\mathbf{x} : \mathbf{x}_0, \mathbf{P}_0)$. Our new sparse

TABLE 3: CRMSE of different filters.

Filter	CKF (5th) $2n^2$ points	GHKF (full) 3^n points	SGHKF (Smolyak) $4n^2 + 1$ points	SGHKF (novel) $2n^2 + 1$ points
CRMSE _{pos} (m)	11.4217	11.4215	15.8146	14.2034
CRMSE _{vel} (m/s)	6.6320	6.6320	14.2974	10.4270
CRMSE _ω (rad)	0.0088	0.0088	0.0207	0.0145

Gauss-Hermite Kalman filter ($m = 3$) with the cubature sets $\{[0], [\sqrt{3}], [\sqrt{3}, \sqrt{3}]\}$ is used in the experiment. In order to make a reliable comparison between different filters, we implement our experiment with the (5th) CKF [28], full tensor-product based Gauss-Hermite Kalman filter ($m = 3$) [2], and Smolyak rule based sparse Gauss-Hermite Kalman filter [18] ($m = 3$).

The above experiment is repeated 100 times independently. Root mean square errors RMSE_{pos}, RMSE_{vel}, and RMSE_ω are used to evaluate the tracking performance of the position, velocity, and the turning rate. The detailed formulations are shown as follows:

$$\begin{aligned}
& \text{RMSE}_{\text{pos}}(k) \\
&= \sqrt{\frac{1}{N_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} \left((\hat{\mathbf{x}}_{1,k}^i - \mathbf{x}_{1,k}^i)^2 + (\hat{\mathbf{x}}_{3,k}^i - \mathbf{x}_{3,k}^i)^2 \right)}, \\
& \text{RMSE}_{\text{vel}}(k) \\
&= \sqrt{\frac{1}{N_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} \left((\hat{\mathbf{x}}_{2,k}^i - \mathbf{x}_{2,k}^i)^2 + (\hat{\mathbf{x}}_{4,k}^i - \mathbf{x}_{4,k}^i)^2 \right)}, \\
& \text{RMSE}_{\omega}(k) = \sqrt{\frac{1}{N_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} (\hat{\mathbf{x}}_{5,k}^i - \mathbf{x}_{5,k}^i)^2}.
\end{aligned} \tag{40}$$

Similarly, CRMSE are used to evaluate the average tracking performance:

$$\begin{aligned}
& \text{CRMSE}_{\text{pos}}(k) \\
&= \frac{1}{N_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} \sqrt{\frac{1}{N_{\text{sim}}} \sum_{j=1}^{N_{\text{sim}}} \left((\hat{\mathbf{x}}_{1,j}^i - \mathbf{x}_{1,j}^i)^2 + (\hat{\mathbf{x}}_{3,j}^i - \mathbf{x}_{3,j}^i)^2 \right)}, \\
& \text{CRMSE}_{\text{vel}}(k) \\
&= \frac{1}{N_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} \sqrt{\frac{1}{N_{\text{sim}}} \sum_{j=1}^{N_{\text{sim}}} \left((\hat{\mathbf{x}}_{2,j}^i - \mathbf{x}_{2,j}^i)^2 + (\hat{\mathbf{x}}_{4,j}^i - \mathbf{x}_{4,j}^i)^2 \right)}, \\
& \text{CRMSE}_{\omega}(k) = \frac{1}{N_{\text{mc}}} \sum_{i=1}^{N_{\text{mc}}} \sqrt{\frac{1}{N_{\text{sim}}} \sum_{j=1}^{N_{\text{sim}}} (\hat{\mathbf{x}}_{5,j}^i - \mathbf{x}_{5,j}^i)^2},
\end{aligned} \tag{41}$$

where $\hat{\mathbf{x}}_{k,j}^i$ represents the k th estimate element of state at time epoch j for the i th experiment. $\mathbf{x}_{k,j}^i$ represents the k th theoretical element of state at time epoch j for the i th experiment. N_{mc} represents the Monte Carlo experiment

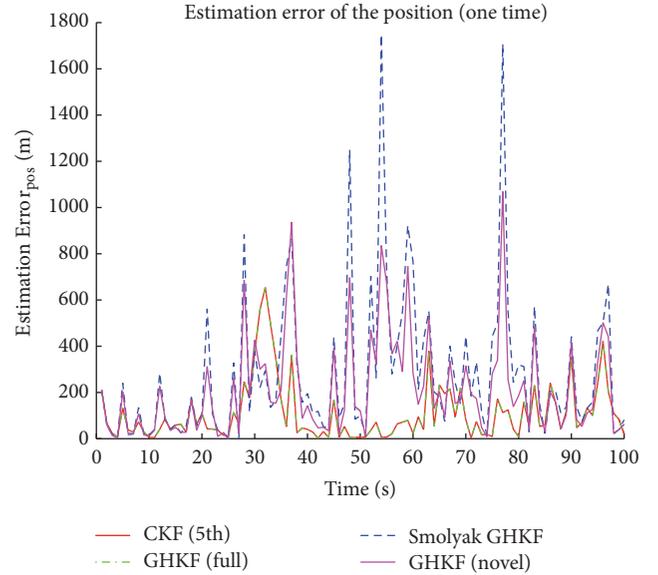


FIGURE 2: Estimation error of position (one simulation).

times. N_{sim} represents the state number of each experiment. We set $N_{\text{mc}} = 100$. $N_{\text{sim}} = 100$ in this experiment. For a single simulation experiment, the tracking performance is shown in Figures 2, 3, and 4. For N_{mc} repeated experiments, the tracking performance is shown in Figures 5, 6, and 7. The CRMSEs of different filters are shown in Table 3.

From Table 3, it can be seen that the full tensor-product based Gauss-Hermite filter and the (5th) CKF have almost exactly the same tracking performance, while the Smolyak-based sparse Gauss-Hermite filter has relatively lower tracking accuracy. The new sparse Gauss-Hermite filter has better tracking accuracy comparing with sparse Gauss-Hermite filter and has less tracking accuracy comparing with the full tensor-product based Gauss-Hermite filter and the (5th) CKF. Our new sparse Gauss-Hermite filter is a good choice to track this manoeuvring target problem with a relative sparse cubature points set where all corresponding weights are kept positive. Furthermore, our method burdens low computational costs and maintains stable numerical accuracy.

6. Conclusion and Discussion

Gauss-Hermite cubature rule is an effective way to approximate nonlinear Gaussian weighted type integral. However, the full tensor-product based Gauss-Hermite cubature rule may cause a dimension explosion and is rarely helpful for

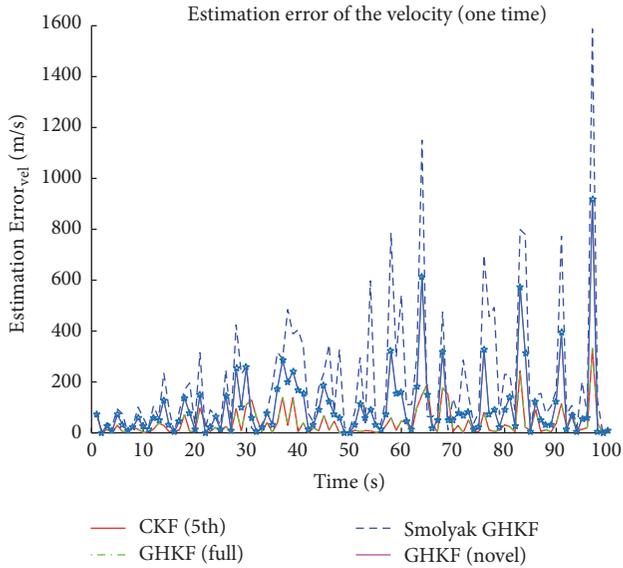


FIGURE 3: Estimation error of velocity (one simulation).

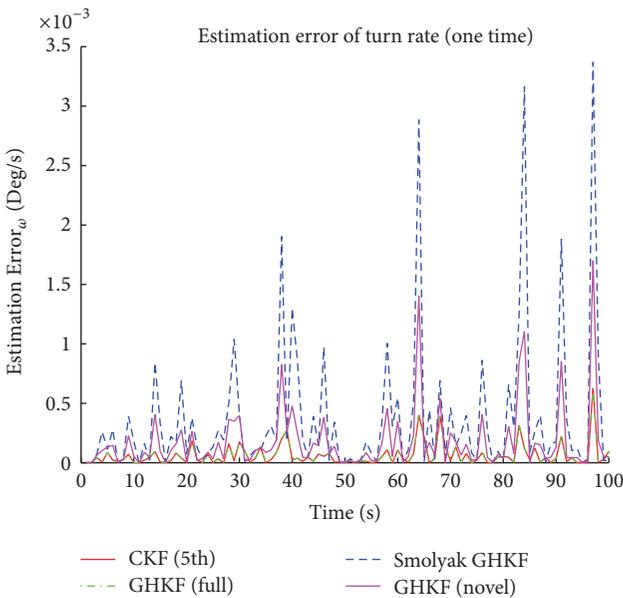


FIGURE 4: Estimation error of turn rate (one simulation).

high dimensional case. Smolyak rule based sparse Gauss-Hermite cubature rule is a way to generate small cubature set by specific generation design, which has a potential drawback that some cubature points may have negative weights. This novel Gauss-Hermite rule is easily generated by using our notation in Definition 2 to classify the full tensor-product based Gauss-Hermite cubature points. Based on a proper relative-ratio-weight criterion, a sparse cubature set is regenerated. Simulations show that our sparse Gauss-Hermite rules will have comparable accuracy with the full

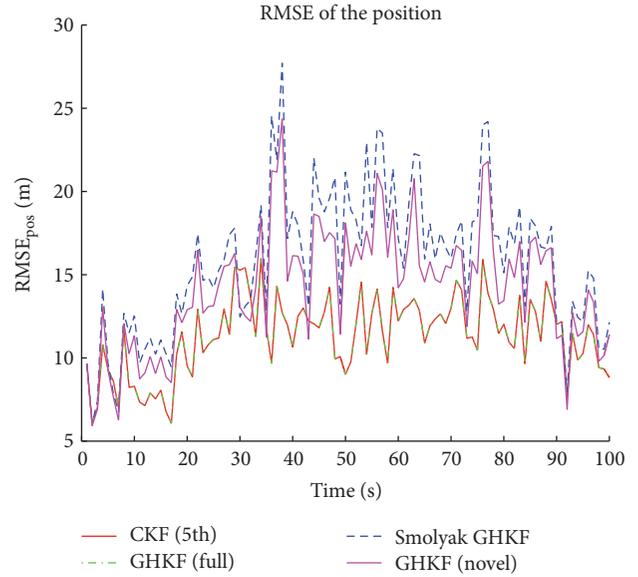


FIGURE 5: RMSEs of position of different filters.

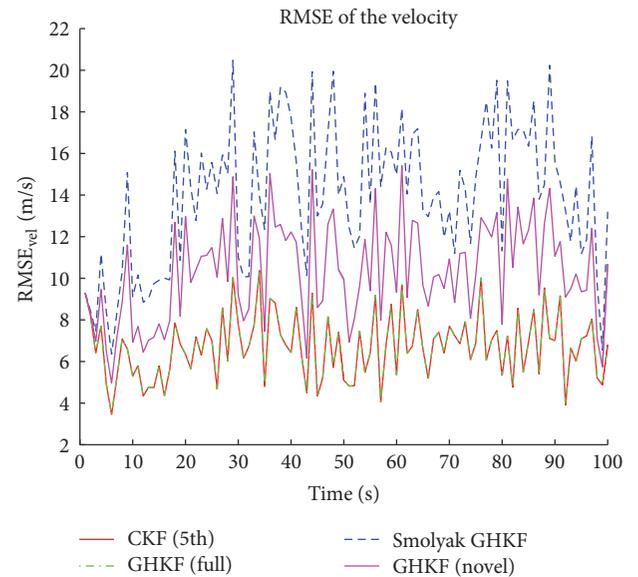


FIGURE 6: RMSEs of velocity of different filters.

tensor-product based Gauss-Hermite filter and higher precision comparing with Smolyak's rule based Gauss-Hermite filter. Furthermore, our method can lead to a substantially reduced number of cubature points and a more stable high dimensional integration. Our method can maintain accuracy of integration at the same time and could be a good choice for practice problems in engineering.

Disclosure

The views and opinions expressed in this paper are those of the authors and do not necessarily reflect the official policy or position of the iGMAS.

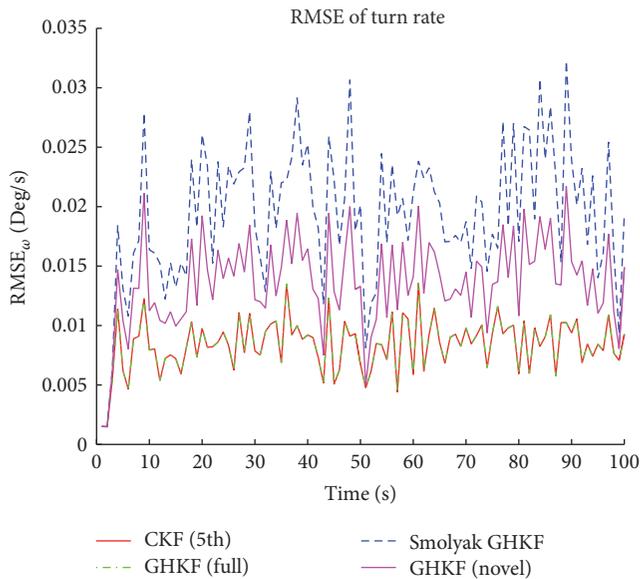


FIGURE 7: RMSEs of turn rate of different filters.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the iGMAS (international GNSS Monitoring and Assessment Service) and the Natural Science Foundation of China (no. 61573367).

References

- [1] F. Gustafsson, F. Gunnarsson, N. Bergman et al., "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.
- [2] I. Arasaratnam, S. Haykin, and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 953–977, 2007.
- [3] M. Šimandl and J. Duník, "Derivative-free estimation methods: new results and performance analysis," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 45, no. 7, pp. 1749–1757, 2009.
- [4] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, 2000.
- [5] Y. Wu, D. Hu, M. Wu, and X. Hu, "A numerical-integration perspective on gaussian filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 8, pp. 2910–2921, 2006.
- [6] A. H. Stroud and D. D. Stancu, "Quadrature formulas with multiple Gaussian nodes," *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis*, vol. 2, no. 1, pp. 129–143, 1965.
- [7] A. Genz, "Fully symmetric interpolatory rules for multiple integrals," *SIAM Journal on Numerical Analysis*, vol. 23, no. 6, pp. 1273–1283, 1986.
- [8] A. Genz and B. D. Keister, "Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight," *Journal of Computational and Applied Mathematics*, vol. 71, no. 2, pp. 299–309, 1996.
- [9] A. Genz, "Fully symmetric interpolatory rules for multiple integrals over hyper-spherical surfaces," *Journal of Computational and Applied Mathematics*, vol. 157, no. 1, pp. 187–195, 2003.
- [10] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *Institute of Electrical and Electronics Engineers. Transactions on Automatic Control*, vol. 54, no. 6, pp. 1254–1269, 2009.
- [11] S. Bhaumik and Swati, "Cubature quadrature Kalman filter," *IET Signal Processing*, vol. 7, no. 7, pp. 533–541, 2013.
- [12] A. K. Singh and S. Bhaumik, "Cubature quadrature Kalman filter for maneuvering target tracking," in *Proceedings of the 1st IEEE International Conference on Smart Sensors and Application, ICSSA 2015*, pp. 24–29, May 2015.
- [13] A. K. Singh and S. Bhaumik, "Higher degree cubature quadrature kalman filter," *International Journal of Control, Automation and Systems*, vol. 13, no. 5, pp. 1097–1105, 2015.
- [14] A. K. Singh, S. Bhaumik, and P. Date, "Quadrature filters for one-step randomly delayed measurements," *Applied Mathematical Modelling. Simulation and Computation for Engineering and Environmental Systems*, vol. 40, no. 19-20, pp. 8296–8308, 2016.
- [15] S. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Soviet Mathematics, Doklady*, vol. 4, no. 5, pp. 240–243, 1963.
- [16] T. Gerstner and M. Griebel, "Dimension-adaptive tensor-product quadrature," *Computing. Archives for Scientific Computing*, vol. 71, no. 1, pp. 65–87, 2003.
- [17] F. Heiss and V. Wünschel, "Likelihood approximation by numerical integration on sparse grids," *Journal of Econometrics*, vol. 144, no. 1, pp. 62–80, 2008.
- [18] B. Jia, M. Xin, and Y. Cheng, "Sparse gauss-hermite quadrature filter with application to spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 2, pp. 367–379, 2011.
- [19] B. Jia, M. Xin, and Y. Cheng, "Sparse-grid quadrature nonlinear filtering," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 48, no. 2, pp. 327–341, 2012.
- [20] R. Radhakrishnan, A. K. Singh, S. Bhaumik, and N. K. Tomar, "Multiple sparse-grid Gauss-Hermite filtering," *Applied Mathematical Modelling. Simulation and Computation for Engineering and Environmental Systems*, vol. 40, no. 7-8, pp. 4441–4450, 2016.
- [21] A. H. Stroud, "Some cubature formulas for the surface of the n -sphere," *SIAM Journal on Numerical Analysis*, vol. 10, no. 4, pp. 559–569, 1973.
- [22] A. Genz and J. Monahan, "Stochastic integration rules for infinite regions," *SIAM Journal on Scientific Computing*, vol. 19, no. 2, pp. 426–439, 1998.
- [23] J. Steinbring and U. D. Hanebeck, "LRKF revisited: the smart sampling Kalman filter (S^2KF)," *Journal of Advances in Information Fusion*, vol. 9, no. 2, pp. 106–123, 2014.
- [24] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1999.
- [25] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.
- [26] G. H. Golub and J. H. Welsch, "Calculation of Gauss quadrature rules," *Mathematics of Computation*, vol. 23, no. 106, article 221, 1969.

- [27] X.-C. Zhang, "A novel cubature kalman filter for nonlinear state estimation," in *Proceedings of the 52nd IEEE Conference on Decision and Control, CDC 2013*, pp. 7797–7802, December 2013.
- [28] B. Jia, M. Xin, and Y. Cheng, "High-degree cubature Kalman filter," *Automatica. A Journal of IFAC, the International Federation of Automatic Control*, vol. 49, no. 2, pp. 510–518, 2013.
- [29] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

