Computationally efficient sparse-grid Gauss-Hermite filtering

Abhinoy Kumar Singh, Rahul Radhakrishnan, Shovan Bhaumik and Paresh Datte

Abstract— **A new nonlinear filtering algorithm based on sparse-grid Gauss-Hermite filter (SGHF) incorporated with the technique of algorithm adapting to dimensions based on their nonlinearity, is presented. The motive of this work is to reduce the computatioanl load of SGHF, while maintaining similar filtering accuracy. This is achieved by implementing adaptive tensor product to construct the multidimensional sparse-grid quadrature points. This reduction in computational burden may increase the scope of application of this filtering algorithm for higher dimensional problems in on-board applications. Performance of the proposed algorithm is illustrated by estimating the frequency and amplitude of multiple superimposed sinusoids.**

I. INTRODUCTION

The problem of non-linear filtering has been widely studied and discussed in literature. This problem involves the recursive estimation of unknown states of a system by making use of noisy measurements. In recent years, the Bayesian framework has been most accepted for addressing these problems. Under this framework, the posterior probability densities are computed using the predicted motion of the unknown states and the measurement likelihood [1]. The discrete state-space model of a dynamic system, which includes both the process and measurement models can be described as

$$x_k = \phi(x_{k-1}) + w_{k-1} \tag{1}$$

and

$$y_k = \gamma(x_k) + v_k \tag{2}$$

respectively. Here $x_k \in \Re^n$ represents the unknown states of the system, $y_k \in \Re^p$ denotes the measurement at any instant k, where $k = 0, 1, 2, ..., N$. $\phi(x_k)$ and $\gamma(x_k)$ are given nonlinear functions. The process and measurement noises are given by $w_k \in \Re^n$ and $v_k \in \Re^p$ respectively. These are assumed to be uncorrelated and normally distributed with zero mean and covariances Q_k and R_k, respectively. The two main steps in Bayesian framework can be described as:

1) Prediction step: In this step, Chapman-Kolmogorov equation is used for obtaining the prior probability density function,

$$p(x_k|y_{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{k-1})dx_{k-1}. \tag{3}$$

2) Update step: In this step, the posterior probability density function is obtained using the prior probability density function and new measurements. The relation is realized with Bayes rule and can be expressed as

$$p(x_k|y_k) = \frac{p(y_k|x_k)p(x_k|y_{k-1})}{\int p(y_k|x_k)p(x_k|y_{k-1})dx_k}. \tag{4}$$

There is no method for obtaining the solution of above equations (3) and (4) when the process and measurement is nonlinear and the pdf encountered is non-Gaussian. This is because of the fact that these integrals become intractable and hence, we have to go for sub-optimal solutions.

Literature about early suboptimal algorithms for nonlinear filtering begins with the extended Kalman filter (EKF) [1]. However, it resulted in undesirable performances like poor tracking accuracy or divergence in estimation error [2]. Later, various nonlinear filtering algorithms such as the unscented Kalman filter (UKF) [2],[3], cubature Kalman filter (CKF) [4] and its variants [5], cubature quadrature Kalman filter (CQKF) [6] and its variants [7] *etc.* were introduced and these filters performed with acceptable accuracy. Square root version of CQKF [8] has also been introduced which takes account of the problem of negative definite covariance matrices, which happens quite often during filtering. To achieve more accuracy, Gauss-Hermite filter (GHF) was also introduced [9], [10]. It makes use of the Gauss-Hermite quadrature rule and has superior performance among all the above mentioned filters. But, it suffers from the *curse of dimensionality* problem since the number of quadrature points required increases exponentially with the increase in dimension of the system. So it is difficult to apply it on-board for higher dimensional problems.

To reduce the computational burden incurred by GHF, multiple quadrature Kalman filter (MQKF) [11] is introduced. It works on the state-space partitioning technique and runs several filters in parallel. This technique significantly reduces the computational load in the algorithm, but its application is limited because of the fact that the function $\phi(x_k)$ must be diagonally occupied so that partitioning can be done. Sparse-grid quadrature filter (SGQF) [12] is an efficient filtering algorithm which can achieve accuracy levels almost as high as GHF, but with comparatively less computational load. It uses the Smolyak rule [13] for extending the one-dimensional quadrature rule to multi-dimensional problems. To further reduce the computational burden, multiple sparse-grid Gauss-Hermite filter (MSGHF) was proposed [14]. Adaptation of SGQF algorithm to nonlinearity has been studied and reported in [15], [16].

In this work, we further reduce the computational burden of SGHF by applying a dimension adaptive tensor product to construct the multidimensional quadratue points. This technique automatically find the dimensions with comparatively less nonlinearity and generate fewer points for approximation along them. Further, by using two separate tuning parameters, a trade-off can be achieved between accuracy and computational burden.

II. DIMENSION ADAPTIVE SPARSE-GRID GAUSS-HERMITE FILTERING

In this section, we propose the dimension adaptive sparse-grid Guass-Hermite filter. As a prelude to this, SGHF is also briefly discussed to convey the idea of how Smolyak rule constructs multidimensional quadrature points for approximation of intractable integrals.

A. *Sparse-grid Gauss-Hermite filter*

Consider an integral of the form

$$I_{n,L}(f^n(x)) = \int_{\Re^n} f^n(x) \aleph(x;0,I_n)dx$$

with an n dimensional function and $\aleph(x;0,I_n)$ denote the normal distribution of x with zero mean and unity covariance. The approximate solution for this integral can be constructed using the Smolyak rule, which uses tensor product of difference formulae computed over a set of accuracy levels defined for univariate quadrature rules. This difference formula for a univariate function can be expressed as

$$\triangle_l f^1(x) = (I_l - I_{l-1}) f^1(x),$$

where $I_0 f^{(1)} = 0$. Now for n dimensional functions, $f^n(x)$, $I_{n,L}(f^n(x))$ can be expressed as

$$I_{n,L}(f^n(x)) = \sum_{\Xi \in \mathbf{N}_q^n} (\triangle_{l_1} \otimes \cdots \otimes \triangle_{l_n}) f^n(x), \qquad (5)$$

where $l_1, l_2, ..., l_n$ represents the univariate accuracy levels corresponding to the dimensions of f^n. N_q^n denotes the set defined by

$$\mathbf{N}_q^n = \left\{ \sum_{j=1}^{n} l_j = n + q \right\} \qquad \text{for} \quad q \geq 0$$
$$= \varnothing \qquad \text{for} \quad q < 0$$

where $\varnothing$ represents the null set and q is an integer *i.e.* $L-n \leq q \leq L-1$ and L the accuracy level corresponding to the approximation of $I_{n,L}(f^n(x))$. This means that the approximation is exact for all the polynomials having degree upto $(2L-1)$. After some algebraic simplifications, (5) can be represented as [12], [17]

$$I_{n,L}(f^n(x)) \approx \sum_{q=L-n}^{L-1} (-1)^{L-1-q} C_{L-1-q}^{n-1} \sum_{\mathbf{N}_q^n} (I_{l_1} \otimes I_{l_2} \otimes ... \otimes I_{l_n}), \qquad (6)$$

where C stands for the binomial coefficient, *i.e.* $C_k^n = n!/k!(n-k)!$. Since I_{l_j} is the single dimensional approximate of integral function over normal distribution, (6) can be rewritten as

$$I_{n,L}(f) \approx \sum_{q=L-n}^{L-1} (-1)^{L-1-q} C_{L-1-q}^{n-1} \sum_{\mathbf{N}_q^n} \sum_{q_{s_1} \in X_{l_1}} \sum_{q_{s_2} \in X_{l_2}} \qquad (7)$$
$$\cdots \sum_{q_{s_n} \in X_{l_n}} f(q_{s_1}, q_{s_2}, ..., q_{s_n}) w_{s_1} w_{s_2} ... w_{s_n},$$

where X_{l_j} is the set of quadrature points for the single dimensional quadrature rule I_{l_j}, $[q_{s_1}, q_{s_2}, ..., q_{s_n}]^T$ is a sparse-grid quadrature point *i.e.* $q_{s_j} \in X_{l_j}$ and w_{s_j} is the weight associated with q_{s_j}. Fig 1 shows the generation of sparse quadrature pints for a 2 dimensional system with accuracy level $L = 3$.

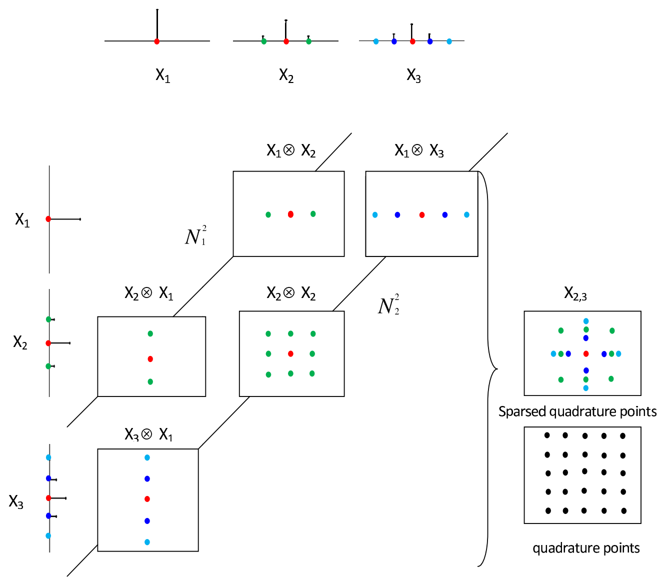


Fig. 1. Generation of sparse-grid Gauss-Hermite points for $n = 2$, $L = 3$

B. *Dimension adaptive sparse-grid Gauss-Hermite filter*

To incur minimum computational burden and maximum integral approximation accuracy, the proposed algorithm adaptively detect dimensions with comparatively lower non-linearity and approximates them with less number of points. This depends on the selection of an admissible index set given a predefined error bound and amount of work involved [17]. From this index set, the corresponding points and weights are calculated.

An index set of dimension n can be defined as $\mathbb{I}_n = \{\lambda : \lambda = (l_1 \quad l_2 \cdots l_n), l_i \in \mathbb{Z}_+\}$, where $\mathbb{Z}_+$ is a set of positive integers. Now, an index set is said to be admissible, if it satisfies

$$\lambda - e_j \in \mathbb{I}_n \quad \forall \quad \lambda \in \mathbb{I}_n, 1 \leq j \leq n, \lambda_j > 1, \qquad (8)$$

where e_j is the j^{th} column of a unit vector and $\lambda \in \mathbb{I}_n$, the indices. The original sparse-grid construction is reformulated using this admissible index set and hence, relation (5) can be rewritten as

$$I_{n,L} f^n(x) = \sum_{\lambda \in \mathbb{I}_n} (\triangle_{\lambda_1} \otimes ... \otimes \triangle_{\lambda_n}) f^n(x). \qquad (9)$$

The main aim of the algorithm is to create an admissible index set ideally so that the accuracy in approximation can be enhanced according to a predefined tolerance level. This gives way for the introduction of g_λ, the local error indicator, which takes care of the error criteria, defined as [17]

$$g_\lambda = max\{\psi \frac{|\triangle_\lambda f|}{|\triangle_1 f|}, (1-\psi)\frac{\varpi_1}{\varpi_\lambda}\}, \qquad (10)$$

where $\psi \in [0,1]$ is called the error weighting parameter and ϖ_λ defines the number of function evaluations or the work involved. Usually, ϖ_1 is taken as unity. This parameter helps in reaching a trade-off between small error and high computational work.

The functioning of the proposed algorithm starts from $\mathbb{I}_n^1 = (1, ..., 1)$, where $\mathbb{I}_n^1$ is the first element of the index set $\mathbb{I}_n$. Then more indices are added successively in such a way that the concerned index set remains admissible and a large error reduction is possibly achieved. The index with the largest error indicator is added to the admissible set and it's forward neighbours are scanned for new admissible indices by computing their error indicators. Forward neighbourhood of an index λ can be defined as $\{\lambda + e_j, 1 \leqslant j \leqslant n\}$, for a maximum number of n indices. The sum of all differential integrals calculated within the admissible set $\mathbb{I}_n$, constitutes the value of the total integral. An index is added to the admissible set immediately after calculating it's error indicator, without even evaluating the error in it's forward neighbour.

Now we go for a partition of the current index set $\mathbb{I}_n$ into active index set $\mathbb{A}$ and old index set $\mathbb{O}$, which are disjoint in nature. $\mathbb{A}$ contains those index sets whose error indicators have been calculated and the error indicators of it's forward neighbours have not been examined yet, whereas, the old index set, $\mathbb{O}$, holds all other indices of $\mathbb{I}_n$ which are not included in $\mathbb{A}$. The error estimate in the approximation process is given by the global error estimate, η. It represents the sum of error indicators present in the active index set $\mathbb{A}$ and can be defined as

$$\eta = \sum_{\mathbb{I}_n^i \in \mathbb{A}} g_\lambda. \tag{11}$$

where $\mathbb{I}_n^i$ is an i^{th} vector representing an index from the index set $\mathbb{I}_n$.

For every iterative step of the algorithm, the index set with the largest error indicator is selected from the active index set and transferred to the old index set and its corresponding error is subtracted from the global error, η. In the meantime, the error indicator of the forward neighbours of that particular index is calculated and transferred to the active index set. For controlling the integration process, a threshold value (TOL) for the error is predefined. Whenever the global error estimate falls below this threshold value, the computation is ceased and the integral value is returned.

III. Simulation Results

A. *Estimation of multiple superimposed sinusoids*

In this problem, we estimate the amplitude and frequency of multiple superimposed sinusoids. Such problems practically appear in many fields like communication systems [18], power systems [11], [19] *etc.*

We consider, the number of sinusoids as three, then the state variable will be $\mathbf{x} = [f_1, f_2, f_3, a_1, a_2, a_3]$, where f_i and a_i are the frequency and amplitude of i^{th} sinusoid. The state is considered to be following random walk model, hence the discrete state space equation will be

$$\mathbf{x}_k = I_6 \mathbf{x}_{k-1} + w_k, \tag{12}$$

where I_6 is a six dimensional unit matrix and w_k is process noise which is added to compensate the uncertainty appeared in sinusoids. The process noise is normally distributed with zero mean and covariance $Q = diag([\sigma_f^2 \ \sigma_f^2 \ \sigma_f^2 \ \sigma_a^2 \ \sigma_a^2 \ \sigma_a^2])$ where σ_f and σ_a are standard deviations for frequency and amplitude.

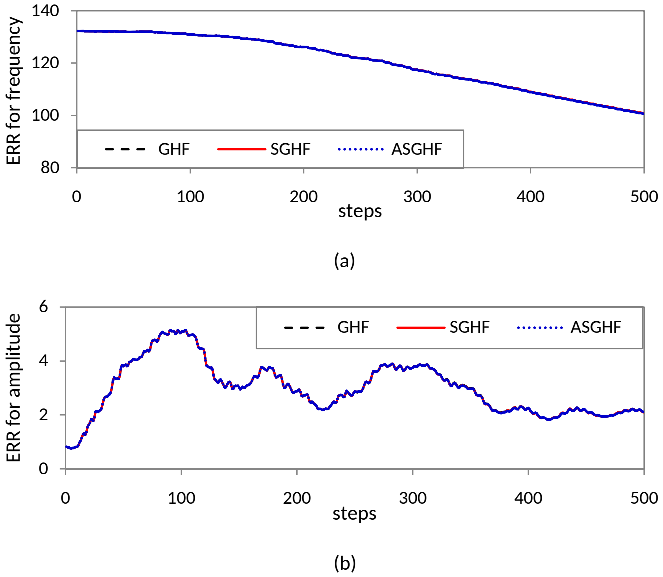
The measurement equation is [11]

$$y_k = \begin{bmatrix} \sum_{j=1}^{3} a_{j,k} cos(2\pi f_{j,k} kT) \\ \sum_{j=1}^{3} a_{j,k} sin(2\pi f_{j,k} kT) \end{bmatrix} + v_k,$$

where v_k is Gaussian noise with zero mean and covariance $R = diag([\sigma_n^2 \ \sigma_n^2])$ with σ_n being the standard deviation for measurement noise. T is the sampling time which is considered as 0.1667 *ms*.

The initial truth and estimates are considered as $[200\ 1000\ 2000\ 5\ 4\ 3]^T$ and $[150\ 900\ 1800\ 4\ 4\ 2]^T$. The initial error covariance and noise covariances, are considered as $\sigma_f^2 = 151 \mu Hz^2/ms^2$, $\sigma_a^2 = 80 \mu V^2/ms^2$, $\sigma_n^2 = 0.09 V^2$, and $P_{0|0} = diag([20^2\ 20^2\ 20^2\ 0.05\ 0.05\ 0.05])$.

For simulation purposes, we use 3-point GHF and 3^{rd}-degree of accuracy level is considered for SGHF (*i.e.* $L = 3$). For ASGHF, the simulation is performed by considering the error weighting parameters as 0.6 and 0.5, while tolerance as 0.53 and 0.6655 for process and measurement equations respectively.



(a)

(b)

Fig. 2. ERR plot for (a) frequency in Hz (b) amplitude in *volt*.

The states are estimated for 500 steps and the results are averaged over 2000 Monte Carlo runs. At each step, a combined error parameter (*ERR*) is evaluated for frequency and amplitude, which is defined as

$$ERR_k = \sqrt{\frac{MSE_{1,k} + MSE_{2,k} + MSE_{3,k}}{3}}, \tag{13}$$

where, for M number of Monte Carlo runs, $MSE_{i,k}$ is

$$MSE_{i,k} = \frac{1}{M} \sum_{j=1}^{M} (\mathbf{x}_{i,k,j} - \hat{\mathbf{x}}_{i,k,j})^2. \tag{14}$$

Filters	Relative comp. time
GHF	1
SGHF	0.17
ASGHF	0.056

TABLE I

RELATIVE COMPUTATIONAL TIME FOR VARIOUS FILTERS

The *ERR* for frequency and amplitude are plotted in Fig 2 using the proposed ASGHF, the SGHF and the GHF. The *ERR* is similar for all the filters and hence it could be concluded that the accuracy of the proposed algorithm is similar to the conventional GHF and SGHF. On the other hand, from the Table I, it could be concluded that the computational burden for the proposed ASGHF is almost 3 times less than the SGHF and 18 times less than the conventional GHF. Due to a similar accuracy at much reduced computational cost, the proposed filter can replace the conventional Gauss-Hermite quadrature based filters.

IV. CONCLUSIONS

A dimension adaptive recursive algorithm for nonlinear filtering based on sparse-grid Gauss-Hermite filter is proposed in this work. It make use of adaptive tensor products which helps in identifying dimensions with less nonlinearity and allocating fewer number of support points for its approximation. This framework helps in maintaining similar accuracy levels when compared to SGHF, with significant reduction in computational load. Apart from the accuracy level *L*, threshold value for error tolerance and error weighting parameters help in fine tuning the filter for high accuracies, while keeping a check on the computational load. Simulation results show that the accuracy of the proposed method is similar to GHF and SGHF, at the expense of very small computational cost. Thus, this filtering method can be considered as an alternative to GHF and SGHF.

REFERENCES

[1] Y. Bar-Shalom, X. R. Li and T. Kirubarajan, Estimation with applications to tracking and navigation, John Wiley & Sons, INC., 2001.

[2] S. Julier, J. Uhlmann and H.F. Durrant-Whyte, A new method for the nonlinear transformation of means and covariances in filters and estimators, IEEE Trans. Autom. Control, vol. 45, no. 3, pp. 477-482, 2000.

[3] S. Julier and J. Uhlmann, Unscented filtering and nonlinear estimation, Proc. of the IEEE, vol. 92, no. 3, pp. 401-422, 2004.

[4] I. Arasaratnam and S. Haykin, Cubature kalman filters, IEEE Tran. Autom. Control, vol. 54, no. 6, pp. 1254-1269, 2009.

[5] I. Arasaratnam, S. Haykin and T. Hurd, Cubature Kalman filters for continuous-discrete systems: theory and simulations, IEEE Trans. Signal Process., vol. 58, no. 10, pp. 4977-4993, 2010.

[6] S. Bhaumik and Swati, Cubature quadrature Kalman filter, IET Signal Process., vol. 7, no. 7, pp. 533-541, 2013.

[7] A. K. Singh and S. Bhaumik, Higher degree cubature quadrature Kalman filter, Int. J. of Control Autom. and syst., vol. 13, no. 5, pp. 1097-1105, 2015.

[8] S. Bhaumik and Swati, Square root cubature quadrature Kalman filter, Asian Journal of Control, vol. 16, no. 2, pp. 617-622, 2014.

[9] K. Ito and K. Xiong, Gaussian filters for nonlinear filtering problems, IEEE Trans. Autom. Control, vol. 45, no. 5, pp. 910-927, 2000.

[10] I. Arasaratnam, S. Haykin and R. Elliott, Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature, Proc. of the IEEE, vol. 95, no. 5, pp. 953-977, 2007.

[11] P. Closas, C.F. Prades and J.V. Valls, Multiple quadrature kalman filtering, IEEE Trans. Signal Process., vol. 60, no. 12, pp. 6125-6137, 2012.

[12] B. Jia, M. Xin and Y. Chen, Sparse-grid quadrature nonlinear filtering, Automatica, vol. 48, no. 2, pp. 327-341, 2012.

[13] S.A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Dokl. Akad. Nauk SSSR, vol. 4, pp. 240-243, 1963.

[14] R. Radhakrishnan, A. K. Singh, S. Bhaumik and N. K. Tomar, Multiple sparse-grid Gauss-Hermite filtering, Appl. Mathematical Modelling, vol. 40, pp. 4441-4450, 2016.

[15] K. Baek and H. Bang, Adaptive sparse grid quadrature filter for spacecraft relative navigation, Acta Astronautica, vol. 87, pp. 96-106, 2013.

[16] T. Sun, M. Xin and B. Jia, Nonlinearity-based adaptive sparse-grid quadrature filter, American Control Conference, July 1-3, 2015.

[17] T. Gerstner and M. Griebel, Dimension-adaptive tensor-product quadrature, Computing, vol. 71, pp. 65-87, 2003.

[18] M. Niedzwiecki and P. Kaczmarek, Estimation and tracking of complex-valued quasi-periodically varying systems, Automatica, vol. 41, no. 9, pp. 1503-1516, 2005.

[19] P.K. Dash and R.K. Mallick, Accurate tracking of harmonic signals in VSC-HVDC systems using PSO based unscented transformation, Int. J. Electr. Power Energy Syst., vol. 33, no. 7, pp. 1315-1325, 2011.