1-1-2012

The Sparse-grid based Nonlinear Filter: Theory and Applications

Bin Jia

THE SPARSE-GRID BASED NONLINEAR FILTER: THEORY AND APPLICATIONS

By

Bin Jia

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Aerospace Engineering
in the Department of Aerospace Engineering

Mississippi State, Mississippi

MAY 2012

THE SPARSE-GRID BASED NONLINEAR FILTER: THEORY AND APPLICATIONS

By

Bin Jia

Approved:

Ming Xin Assistant Professor of Aerospace Engineering (Major Professor)	Yang Cheng Assistant Professor of Aerospace Engineering (Committee Member)
Jenny Q. Du Associate Professor of Electrical and Computer Engineering (Committee Member)	Masoud Rais-Rohani Professor of Aerospace Engineering (Committee Member)
Randolph F. Follett Assistant Professor of Electrical and Computer Engineering (Committee Member)	J. Mark Janus Associate Professor of Aerospace Engineering (Graduate Coordinator)
Sarah A. Rajala Dean of James Worth Bagley College of Engineering	

Name: Bin Jia

Date of Degree: May 11, 2012

Institution: Mississippi State University

Major Field: Aerospace Engineering

Major Professor: Dr. Ming Xin

Title of Study: THE SPARSE-GRID BASED NONLINEAR FILTER: THEORY AND APPLICATIONS

Pages in Study: 144

Candidate for Degree of Doctor of Philosophy

Filtering or estimation is of great importance to virtually all disciplines of engineering and science that need inference, learning, information fusion, and knowledge discovery of dynamical systems. The filtering problem is to recursively determine the states and/or parameters of a dynamical system from a sequence of noisy measurements made on the system. The theory and practice of optimal estimation of linear Gaussian dynamical systems have been well established and successful, but optimal estimation of nonlinear and non-Gaussian dynamical systems is much more challenging and in general requires solving partial differential equations and intractable high-dimensional integrations. Hence, Gaussian approximation filters are widely used.

In this dissertation, three innovative point-based Gaussian approximation filters including sparse Gauss-Hermite quadrature filter, sparse-grid quadrature filter, and the anisotropic sparse-grid quadrature filter are proposed. The relationship between the proposed filters and conventional Gaussian approximation filters is analyzed. In particular, it is proven that the popular unscented Kalman filter and the cubature Kalman filter are subset of the proposed sparse-grid filters. The sparse-grid filters are employed in

three aerospace applications including spacecraft attitude estimation, orbit determination, and relative navigation. The results show that the proposed filters can achieve better estimation accuracy than the conventional Gaussian approximation filters, such as the extended Kalman filter, the cubature Kalman filter, the unscented Kalman filter, and is computationally more efficient than the Gauss-Hermite quadrature filter.

DEDICATION

To

My parents and my wife

ACKNOWLEDGEMENTS

I want to thank my advisor Dr. Ming Xin who gives me financial support and research guide to finish my Ph. D dissertation. In addition, I want to thank Dr. Cheng Yang who gives me a lot of help. Furthermore, I want to thank my school and department that provide me a comfortable environment to finish my Ph. D. degree. I also want to thank my committee for their comments on this dissertation.

TABLE OF CONTENTS

Page

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF TABLES vii

LIST OF FIGURES viii

LIST OF ALGORITHEMS x

CHAPTER

I. INTRODUCTION 1

1.1 Overview 1
1.2 Contributions of The Dissertation 4
1.2.1 Three New Filters Based On The Sparse-grid Method 4
1.2.2 Relationship Between The Sparse-grid Based Filters and Other Gaussian Approximation Filters 5
1.2.3 Space Applications 5
1.3 Organization of The Dissertation 5

II. BACKGROUND AND LITERATURE REVIEW 7

2.1 Optimal Bayesian Filtering 7
2.1.1 Continuous-discrete system 7
2.1.2 Discrete System 8
2.2 Approximate Nonlinear Filters 10
2.2.1 Gaussian Approximation Filters 10
2.2.1.1 Extended Kalman Filter 11
2.2.1.2 Point-based Gaussian approximation filters 12
2.2.1.3 Unscented Transformation 16
2.2.1.4 Central Difference 17
2.2.1.5 Gauss-Hermite Quadrature 18
2.2.1.6 Cubature Rules 19
2.2.1.7 Divided Difference Filters 22
2.2.1.7.1 The First Order Stirling‟s Interpolation 22

2.2.1.7.2 The Second Order Stirling"s Interpolation....23
2.2.1.8 Discussions....23
2.2.2 Non-Gaussian Approximation Filters....24
2.2.2.1 Sequential Monte Carlo Method....25
2.2.2.2 Point-Mass Filter....29
2.2.2.3 Gauss Sum Filter....30

III. SPARSE-GRID BASED NONLINEAR FILTERS....32

3.1 Sparse Gauss-Hermite Quadrature Filter....33
3.1.1 Sparse Gauss-Hermite Quadrature....33
3.2 Sparse-grid Quadrature Filter....49
3.2.1 Univariate Quadrature Points and Weights....50
3.2.2 Multi-dimensional Quadrature Points and Weights....54
3.2.3 Comparison of the sparse-grid quadrature rules, the unscented transformation, and the Gauss-Hermite quadrature rule....58
3.2.4 The relationship between the sparse-grid quadrautre filters and the cubature Kalman filters....63
3.3 Anisotropic Sparse-grid Quadrature Nonlinear Filter....72
3.3.1 Anisotropic Sparse-grid Quadrature....72
3.3.2 Analysis of the Accuracy of the Anisotropic Sparse-grid Quadrature....76
3.3.3 The Anisotropic Sparse-grid Quadrature Filter....83

IV. AEROSPACE APPLICATIONS....85

4.1 Spacecraft Attitude Estimation....85
4.1.1 Attitude Kinematics Model....86
4.1.2 Vector Observation Model....87
4.1.3 Sparse Gauss-Hermite Quadrature Filter for Attitude Estimation....88
4.1.3.1 Prediction Step....88
4.1.3.2 Update Step....91
4.1.4 Numerical Results and Analysis....94
4.1.4.1 The Performance of SGHQF....94
4.1.4.2 The Performance of ASGHQF....103
4.2 Spacecraft Orbit Determination....107
1.1.1 The First Scenario: Small Initial Estimation Errors....112
4.2.2 The Second Scenario: Large Initial Estimation Errors....114
4.2.2.1 Results of UKFs and level-2 SGQFs....114
4.2.2.2 Results of level-3 SGQFs....117
4.3 Spacecraft Relative Navigation....120
4.3.1 Relative Navigation Model....122
4.3.1.1 Relative Attitude Kinematics....122
4.3.1.2 Relative Orbit Dynamic Equations....123

4.3.1.3 Vision-Based Measurement Model....124
4.3.2 Relative Attitude And Orbit Estimation....125
4.3.2.1 Relative Navigation Algorithm....125
4.3.2.2 Approximation of The Process Noise....127
4.3.3 Simulation Results and Analysis....128

V. CONCLUSIONS AND FUTURE RESEARCH....134

REFERENCES....136

LIST OF TABLES

TABLE Page

3.1 Number of SGHQ points for different univariate GHQ point-selection strategies with the accuracy level-2 and level-3 44

3.2 Univariate GHQ points and weights of the first case 46

3.3 Univariate GHQ points and weights of the second case 47

4.1 Total number of points for 6-dimensional SGHQFs (level-2 and level-3), CKF, UKF, and GHQF 95

4.2 Importance vector $\boldsymbol{\alpha}$ of ASGHQF and numbers of points of different filters 104

4.3 Parameters of SGQFs 111

4.4 Number of Points for different SGQFs and GHQF 119

4.5 Locations of beacons 129

LIST OF FIGURES

FIGURE		Page
2.1	Bayesian filtering framework	9
3.1	Multivariate GHQ and SGHQ with the accuracy level-3 for 2-dimensional problems	36
3.2	Point set illustration for 2-dimensional SGHQ (level-2 and level-3)	45
3.3	Point set illustration for 3-dimensional SGHQ (level-2 and level-3)	45
4.1	Norm of attitude estimation errors for SGHQFs using different point-selection strategies and accuracy levels (Case 2)	97
4.2	Norm of attitude estimation errors in comparison with EKF, UKF, CKF and GHQF (Case 2)	98
4.3	Attitude errors of SGHQFs (level-2), UKF, CKF, and EKF with 3 σ -error bounds (Case 3)	99
4.4	Attitude errors of SGHQFs (level3), UKF, and CKF with 3 σ -error bounds (Case 3)	100
4.5	Norm of attitude estimation errors for SGHQFs using different point-selection strategies and accuracy levels (Case 3)	102
4.6	Norm of attitude estimation errors (Case 3)	102
4.7	Attitude estimation errors of EKF, UKF, CKF, and GHQF	106
4.8	Attitude estimation errors of EKF, ASGHQFs, and GHQF	106
4.9	Comparison of total attitude estimation error norms using EKF, UKF, CKF, ASGHQFs and GHQF	107
4.10	Illustration of the observation geometry	109
4.11	RMSEs of the position with small initial estimation errors	113

4.12 RMSEs of the velocity with small initial estimation errors 113

4.13 RMSEs of the position with large initial estimation errors: EKF, UKFs, and GHQF 114

4.14 RMSEs of the velocity with large initial estimation errors: EKF, UKFs, and GHQF 115

4.15 RMSEs of the position with large initial estimation errors: EKF, level-2 SGQFs, and GHQF 116

4.16 RMSEs of the velocity with large initial estimation errors: EKF, level-2 SGQFs, and GHQF 117

4.17 RMSEs of the position with large initial estimation errors: EKF, UKF, level-3 SGQFs, and GHQF 118

4.18 RMSEs of the velocity with large initial estimation errors: EKF, UKF, level-3 SGQFs, and GHQF 119

4.19 Averaged absolute attitude error of EKF, SGHQFs, UKF, and CKF 131

4.20 Averaged absolute error of gyro biases of EKF, SGHQFs, UKF, and CKF for the chief spacecraft 131

4.21 Averaged absolute error of gyro biases of EKF, SGHQFs, UKF, and CKF for the deputy spacecraft 132

4.22 Averaged absolute error of relative position of SGHQFs and CKF 132

4.23 Averaged absolute error of relative velocity of SGHQFs and CKF.....133

LIST OF ALGORITHEMS

ALGORITHEM Page

3.1 Generate SGHQ Points and Weights 38

4.1 SGHQF for Spacecraft Attitude Estimation.....93

CHAPTER I

INTRODUCTION

1.1 Overview

Nonlinear estimation/filtering has attracted a great deal of attention in the past few decades because it has a broad range of applications in science and engineering [1-18], such as communication[1] ,target detection and tracking [2-6, 7 , 8], ocean acoustics[10], location[9], SAR motion compensation [15], and fault detection and correction [16-18]. The general filtering problem can be addressed using Bayesian estimation theory [19]. In the Bayesian framework, the probability density function (pdf) of states between measurements for continuous dynamic systems can be propagated by solving the Fokker-Planck-Kolmogorov equation (FPKE) [20-24] and the pdf of the states can be updated by the Bayesian rule. For discrete dynamic systems, the FPKE will be replaced by the Chapman-Kolmogorov equation [25]. Because the Chapman-Kolmogorov equation requires integration with respect to the pdf, which has no closed-form solution for general nonlinear and non-Gaussian systems, numerous approximation methods have been proposed, such as the sequential Monte-Carlo method [26], the point-mass method [27, 28], and the Gaussian mixture method [3, 5, 29-31]. For general nonlinear, non-Gaussian filters, although the sequential Monte Carlo method (particle filters) can successfully solve the integration problem, the update of the particle weights often leads to the degeneracy problem[32]. In addition, the number of particles of the particle filters increases rapidly with the increase of the system dimension. The point-mass method also

suffers the same problem of computational complexity as the sequential Monte Carlo method. For the Gaussian mixture method, the number of Gaussian components is difficult to control and the computational burden is large. For nonlinear and Gaussian systems, the Gaussian approximation filters can be derived. They are more computationally efficient than the non-Gaussian filters. In addition, Gaussian approximation filters can be used in Gaussian mixture filter [33] and sequential Monte-Carlo method [34]. Hence, in this dissertation, we mainly focus on the Gaussian approximation filters.

There are many Gaussian approximation filters that use different numerical methods to calculate the integrals in the Bayesian filtering. The typical integral methods include unscented transformation [35, 36], cubature rules [37-40], and Gauss-Hermite quadrature rule [29]. Among the Gaussian approximation filtering techniques, Gauss-Hermite quadrature filter (GHQF) has been proven to be the most accurate in solving estimation problems when the state and noise distributions are Gaussian. The conventional GHQF uses tensor product rule to extend one dimensional Gauss-Hermite quadrature rule to the multi-dimensional problem. The drawback of this method is that the number of points increases exponentially with the increase of dimension. Hence, it is difficult to use it in high dimensional problems. To alleviate the curse of dimensionality problem generated by the product rule, the sparse-grid method has been proposed and widely used. The original idea of the sparse-grid method can be traced back to Russian mathematician Smolyak [41], who used a special method to choose points (sparse-grid) and made the number of necessary points significantly less than that of using the direct product rule. As a result, the computational cost does not increase exponentially by using the sparse-grid method. Although the sparse-grid has been used in the numerical

integration context, the application to nonlinear filtering is new. In this dissertation, three sparse-grid based filters are proposed. It will be shown that they can achieve more accurate results than other Gaussian approximation filters with moderate computation complexity.

The proposed sparse-grid based filters are applied to three important space applications: spacecraft attitude estimation, orbit determination, and the relative navigation between two spacecraft. Many nonlinear filtering methods, such as the extended Kalman filter (EKF) [19, 42], the unscented Kalman filter (UKF) [35, 36], and the particle filter (PF) [32, 43-46] have been employed for spacecraft attitude estimation, since it is a nonlinear filtering problem. EKF is the most widely used nonlinear filtering method for spacecraft attitude estimation [47]. A simplified Kalman filter and smoother for spacecraft attitude estimation based on the QUEST algorithm was proposed in [48]. A more robust approach named extended quaternion-estimator based on the EKF and quadratic constrained programming was proposed in [49]. The UKF [50] has demonstrated more accurate and robust performance than the EKF in attitude estimation when the initial attitude estimation error is large. The PF has been shown to achieve better accuracy than the UKF and the EKF at the expense of high computational complexity [43, 44, 46]. Similarly, EKF has been widely used in orbit determination as well [51]. Many other nonlinear filtering algorithms such as UKF [52] and PF [53] have also been investigated to improve the orbit determination performance. In this dissertation, it can be shown that the proposed sparse-grid based filters can achieve higher order accuracy than UKF, and UKF is shown to be a subset of the sparse-grid based filter. The spacecraft attitude estimation and orbit determination problems will be used to demonstrate the effectiveness of the sparse-grid filters and compare the

performance of other Gaussian approximation filters. To show the performance of the sparse-grid filters for high-dimensional problems, they are applied to the relative navigation problem. Relative navigation is one of the key technologies for the satellite formation flying [54, 55]. Many GPS-like missions that need spacecraft formation require information of relative attitude and position between spacecraft. It is a high dimensional problem that is more challenging than attitude estimation and orbit determination. Thus, it is a good test bed to evaluate the estimation performance and the computational efficiency of the sparse-grid filters.

1.2 Contributions of The Dissertation

1.2.1 Three New Filters Based On The Sparse-grid Method

Three new filters based on the sparse-grid theory are proposed. The first filter is the Gauss-Hermite quadrature filter (GHQF) that uses the Gauss-Hermite quadrature rule to generate the univariate Gauss-Hermite quadrature (GHQ) points and weights, and then uses the sparse-grid method to extend them to the multi-dimensional GHQ point set and weights. The second filter is the general sparse-grid quadrature filter (SGQF) that utilizes the moment matching method to generate the univariate quadrature points and weights and then uses the sparse-grid method to extend them to the multi-dimensional quadrature point set and weights. To further improve computation efficiency, the third filter, named the anisotropic sparse-grid filter, is proposed to reduce the number of sparse-grid points based on system dynamics and uncertainty information such that fewer points can be placed along less important dimensions.

1.2.2 Relationship Between The Sparse-grid Based Filters and Other Gaussian Approximation Filters

The relationship between the sparse-grid based filters, sigma-point based filters (UKF and CKF), and the Gauss-Hermite quadrature filter are rigorously analyzed. It can be revealed that the UKF and CKF are a subset of the sparse-grid filters at the accuracy level-2 and thus the sparse-grid filters can achieve higher accuracy level than the sigma-point filters. In addition, the accuracy and computational complexity are quantitatively analyzed for all sparse-grid based filters.

1.2.3 Space Applications

The proposed sparse-grid filters are applied to three important space applications, including orbit determination, attitude estimation, and relative navigation. The performance of the new proposed filters are shown to be better than the conventional Gaussian approximation filters, such as EKF, UKF, CKF in terms of estimation accuracy, and GHQF in terms of computation efficiency.

1.3 Organization of The Dissertation

The dissertation is organized as follows: Chapter 2 briefly reviews the literature of nonlinear filtering methods. The theory of Bayesian filtering and popular Gaussian approximation filters such as the extended Kalman filter, the unscented Kalman filter, the cubature Kalman filter, the Gauss-Hermite quadrature filter, and the central difference Kalman filter, are reviewed.

In Chapter 3, three sparse-grid based filters are proposed. Both conventional and anisotropic sparse-grid theories are introduced. Theoretical analysis of the relationship between the conventional sparse-grid quadrature, anisotropic sparse-grid quadrature, unscented transformation, cubature rule, and Gauss-Hermite quadrature filters is given.

In Chapter 4, the three proposed filters are applied in solving three space estimation problems: orbit determination, attitude estimation, and relative navigation. The simulation results show the proposed filters are better than EKF, UKF and CKF, they are computationally more efficient than GHQF.

Chapter 5 reviews the main contributions of this dissertation and gives some future research directions.

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

In this chapter, the Bayesian filtering framework as well as its variants is reviewed.

2.1 Optimal Bayesian Filtering

The mission of the nonlinear filtering is to estimate hidden variables or states of a system from noise corrupted observations. Roughly speaking, there are two categories of nonlinear filtering methods in terms of the method of data processing. One method is batch processing and the other is recursive processing. Here, we mainly focus on recursive nonlinear filtering.

2.1.1 Continuous-discrete system

In many systems, the measurements are obtained discretely from continuous-time processes. The dynamics of the system is governed by a stochastic differential equation

$$d\mathbf{x}(t) = \boldsymbol{f}\left(\mathbf{x}(t), t\right)dt + \sqrt{\mathbf{Q}}d\boldsymbol{\beta}(t) \tag{2.1}$$

where $\mathbf{x}(t)$ is the state of the system at the time t; $\boldsymbol{f}$ is the nonlinear drift function; $\mathbf{Q}$ is the spectral density matrix; and $\boldsymbol{\beta}$ is the standard Brownian motion.

The noisy observations are obtained discretely at time instant k

$$\mathbf{y}_k = \boldsymbol{h}\left(\mathbf{x}_k\right) + \mathbf{n}_k \tag{2.2}$$

where $\mathbf{y}_k$ are the noisy observations, $\boldsymbol{h}$ is the measurement equation, and $\mathbf{n}_k$ is the measurement noise.

By Bayesian filtering theory [19, 42], there are two steps to obtain the estimated states.

1. *Prediction step*: predict the pdf between the measurement intervals.
2. *Update step*: update the pdf using the corrupted measurements.

The prediction of the pdf from time $k-1$ to time k can be obtained by solving the well-known Fokker-Planck-Kolmogorov equation.

$$\frac{\partial p}{\partial t} = -\boldsymbol{f}^T \frac{\partial p}{\partial \mathbf{x}_t} - \mathrm{tr}\left(\frac{\partial \boldsymbol{f}}{\partial \mathbf{x}_t}\right) p + \frac{1}{2}\mathrm{tr}\left(\mathbf{Q}\frac{\partial^2 p}{\partial t^2}\right) \tag{2.3}$$

where $p = p\left(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}\right)$ with $\mathbf{y}_{1:k-1}$ denoting the measurement sequences up to time *k*-1; „tr" denotes the trace operator.

When the measurement $\mathbf{y}_k$ is available, a posterior pdf is updated by the Bayesian rule.

$$p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k}\right) = \frac{p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}\right) p\left(\mathbf{y}_k \mid \mathbf{x}_k\right)}{\int p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}\right) p\left(\mathbf{y}_k \mid \mathbf{x}_k\right) d\mathbf{x}_k} \tag{2.4}$$

where the predicted $p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}\right)$ can be obtained from Eq. (2.3).

2.1.2 Discrete System

Assume the state-space description of a discrete dynamical system is

$$\mathbf{x}_k = \boldsymbol{f}\left(\mathbf{x}_{k-1}\right) + \mathbf{v}_{k-1} \tag{2.5}$$

$$\mathbf{y}_k = \boldsymbol{h}\left(\mathbf{x}_k\right) + \mathbf{n}_k \tag{2.6}$$

where $\boldsymbol{f}$ and $\boldsymbol{h}$ are nonlinear functions. Eq. (2.5) is called dynamic equation, while Eq. (2.6) is called the measurement equation. $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are white Gaussian process noise and measurement noise, respectively, with $\mathbf{v}_{k-1} \sim N\left(\mathbf{0}, \mathbf{Q}_{k-1}\right)$ and $\mathbf{n}_k \sim N\left(\mathbf{0}, \mathbf{R}_k\right)$. The filtering problem is estimating the system state $\mathbf{x}_k$ from the noisy measurement $\mathbf{y}_k$.

The optimal estimated state $\hat{\mathbf{x}}_k$ can be obtained by the recursive Bayesian filtering algorithm. The Bayesian filter is a general probabilistic approach to estimate the pdf recursively from noisy or corrupted observations. Given the pdf $p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})$ at time $k-1$, the predicted conditional pdf $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})$ satisfies the Chapman-Kolmogoroff equation

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1} \tag{2.7}$$

When the measurement at time k is available, a posterior conditional pdf can be obtained by the Bayesian rule

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) = \frac{p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) p(\mathbf{y}_k \mid \mathbf{x}_k)}{\int p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) p(\mathbf{y}_k \mid \mathbf{x}_k) d\mathbf{x}_k} \tag{2.8}$$

Equation (2.7) is the prediction formula, while Eq. (2.8) is the update formula. The prediction step is to obtain the pdf evolution from time $k-1$ to k and the update step gives a posterior pdf at time k. The moments can be computed from this a posterior pdf.

A nonlinear filtering algorithm described above can be represented in the block diagram as shown in Figure 2.1.

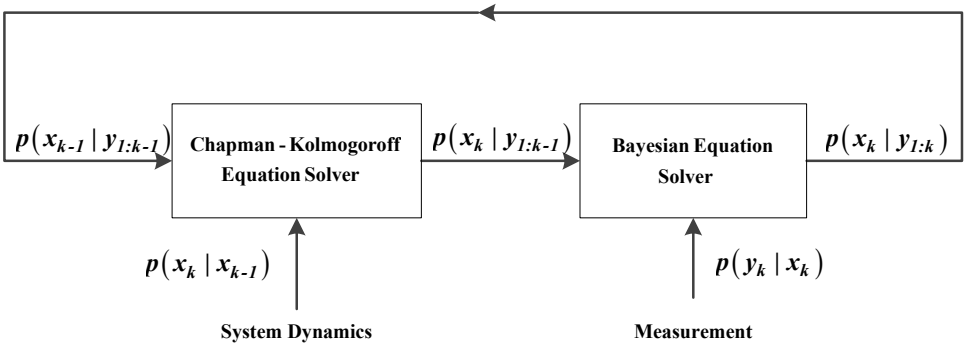

Figure 2.1 Bayesian filtering framework

Initially, $p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1})$ is assumed to be known at time $k-1$, and then $p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$ can be computed by Eq. (2.7). When the measurement arrives at time k, a posterior pdf $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ at time k is calculated by the Bayesian rule. Then the estimated states $\hat{\mathbf{x}}_k$ at time k can be calculated from the corresponding pdf.

$$\hat{\mathbf{x}}_k = \int \mathbf{x}_k p(\mathbf{x}_k | \mathbf{y}_{1:k}) d\mathbf{x}_k \tag{2.9}$$

State estimation can be recursively calculated following this procedure.

Although the Bayesian filter has a simple form, it is difficult to realize. The recursive propagation of a posterior density given by (2.7) and (2.8) is only a conceptual solution in the sense that it cannot be determined analytically in general. Thus, one has to use approximations or suboptimal Bayesian algorithms.

2.2 Approximate Nonlinear Filters

The celebrated Kalman filter is the optimal solution to Bayesian filtering problem when the system model is linear and the noise distribution is Gaussian. When the system model is nonlinear or the noise distributions are not Gaussian, various numerical methods have been proposed to calculate the integrals in Eq. (2.7) and Eq. (2.8). We briefly summarize them into the following two categories: Gaussian approximation filters and non-Gaussian approximation filters.

2.2.1 Gaussian Approximation Filters

This kind of filters assumes that all the models and related pdfs are Gaussian. These methods include the extended Kalman filter (EKF) [42], the unscented Kalman filter (UKF) [35, 36], the central difference filter (CDF), the divided difference filter (DDF), and the Gauss-Hermite quadrature filter (GHQF). Among all Gaussian approximation filters, the EKF is the most widely used. The EKF linearizes the nonlinear

model of Eqs. (2.5) and (2.6) using the first-order truncated Taylor series expansion around the current estimate in order to use the Kalman filter framework for the Gaussian nonlinear system. The central difference filter [56] adopts an alternative linearization method called the central difference approximation to approximate the derivative. UKF and GHQF are quadrature point based methods (unscented transformation and Gauss-Hermite quadrature, respectively) to approximate the optimal Bayesian integrals in Eq. (2.7) and Eq. (2.8). Compared with the EKF, the UKF and GHQF are more accurate, since they consider higher-order terms in the nonlinear system model.

2.2.1.1 Extended Kalman Filter

The extended Kalman filtering algorithm can be also described by the prediction and update steps.

The predicted state and covariance can be approximated by [42]

$$\hat{\mathbf{x}}_{k|k-1} = \boldsymbol{f}\left(\hat{\mathbf{x}}_{k-1|k-1}\right) \tag{2.10}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^{T} + \mathbf{Q}_{k-1} \tag{2.11}$$

where $\hat{\mathbf{x}}_{k-1|k-1}$ is the estimated state and $\mathbf{F}_{k-1}$ is the Jacobian matrix of the function $\boldsymbol{f}$ evaluated at $\hat{\mathbf{x}}_{k-1|k-1}$.

The updated state and covariance are given by [42]

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{L}_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_k\right) \tag{2.12}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{L}_k\mathbf{P}_{yy}\mathbf{P}_{k|k-1}^{T} \tag{2.13}$$

where the predicted observation $\hat{\mathbf{y}}_k$ is given by

$$\hat{\mathbf{y}}_k = \boldsymbol{h}\left(\hat{\mathbf{x}}_k\right) \tag{2.14}$$

and

$$\mathbf{P}_{yy} = \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{T} + \mathbf{R}_k \tag{2.15}$$

Note, $\mathbf{H}_k$ is the Jacobian matrix of the function $\boldsymbol{h}$ evaluated at $\hat{\mathbf{x}}_{k|k-1}$. The Kalman gain $\mathbf{L}_k$ is given by [42]

$$\mathbf{L}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k\right)^{-1} \tag{2.16}$$

2.2.1.2 Point-based Gaussian approximation filters

Point-based Gaussian approximation filters use some points and weights according to certain numerical rules such as Gauss-Hermite quadrature (GHQ) rule, unscented transformation (UT), and cubature rule, to approximate the integrals in the Bayesian filtering algorithm. Under the assumption of Gaussian distributions, Eq. (2.7) and (2.8) can be rewritten as

$$p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}\right) = \int_{\mathbb{R}^n} \frac{1}{(2\pi)^{n/2}\left(\det \mathbf{Q}_{k-1}\right)^{1/2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}_k - f\left(\mathbf{x}_{k-1}\right)\right)^T \mathbf{Q}_{k-1}^{-1}\left(\mathbf{x}_k - f\left(\mathbf{x}_{k-1}\right)\right)\right) \times p\left(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}\right) d\mathbf{x}_{k-1} \tag{2.17}$$

$$p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k}\right) = c\exp\left(-\frac{1}{2}\left(\mathbf{y}_k - h(\mathbf{x}_k)\right)^T \mathbf{R}_k^{-1}\left(\mathbf{y}_k - h(\mathbf{x}_k)\right)\right) p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}\right) \tag{2.18}$$

where c is a normalization constant. Assume that the pdf of the state is also Gaussian and thus only the mean and covariance need to be calculated.

After some derivation, one can arrive at the following Gaussian approximation filtering algorithm [29, 35, 36, 57]:

$$\hat{\mathbf{x}}_{k|k-1} = \int_{\mathbb{R}^n} \boldsymbol{f}\left(\mathbf{x}_{k-1}\right) N\left(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1|k-1}\right) d\mathbf{x}_{k-1} \tag{2.19}$$

$$\mathbf{P}_{k|k-1} = \int_{\mathbb{R}^n} \boldsymbol{f}\left(\mathbf{x}_{k-1}\right) f\left(\mathbf{x}_{k-1}\right)^T N\left(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1|k-1}\right) d\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k|k-1}\hat{\mathbf{x}}_{k|k-1}^T + \mathbf{Q}_{k-1} \tag{2.20}$$

where $N\left(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1|k-1}\right)$ denotes the multivariate normal distribution with mean $\hat{\mathbf{x}}_{k-1}$ and covariance $\mathbf{P}_{k-1|k-1}$.

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{L}_k\left(\mathbf{y}_k - \hat{\mathbf{y}}_k\right) \tag{2.21}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{L}_k\mathbf{P}_{xy}^T \tag{2.22}$$

where $$\mathbf{L}_k = \mathbf{P}_{xy}\left(\mathbf{R}_k + \mathbf{P}_{yy}\right)^{-1} \tag{2.23}$$

$$\hat{\mathbf{y}}_k = \int_{\mathbb{R}^n} \boldsymbol{h}(\mathbf{x}_k) N\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right) d\mathbf{x}_k \tag{2.24}$$

$$\mathbf{P}_{xy} = \int_{\mathbb{R}^n} \left(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{h}(\mathbf{x}_k) - \hat{\mathbf{y}}_k\right)^T N\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right) d\mathbf{x}_k \tag{2.25}$$

$$\mathbf{P}_{yy} = \int_{\mathbb{R}^n} \left(\boldsymbol{h}(\mathbf{x}_k) - \hat{\mathbf{y}}_k\right)\left(\boldsymbol{h}(\mathbf{x}_k) - \hat{\mathbf{y}}_k\right)^T N\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}\right) d\mathbf{x}_k \tag{2.26}$$

The integrals in Eqs. (2.19)-(2.20) and Eqs. (2.24)-(2.26) can be approximated by the GHQ, UT, or cubature rule. They can be approximated by the sparse Gauss-Hermite quadrature (SGHQ) rule as well, which will be presented in the next chapter.

The following Gaussian-type integral can be approximated by the quadrature.

$$\int_{\mathbb{R}^n} \boldsymbol{g}(\mathbf{x}) N(\mathbf{x}; \mathbf{0}, \mathbf{I}) d\mathbf{x} \approx \sum_{i=1}^{N_p} W_i \boldsymbol{g}(\boldsymbol{\gamma}_i) \tag{2.27}$$

where N_p is the total number of points; W_i and $\boldsymbol{\gamma}_i$ are quadrature weights and quadrature points, respectively.

A more general Gaussian-type integral $\int_{\mathbb{R}^n} \boldsymbol{g}(\mathbf{x}) N(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}) d\mathbf{x}$ can be approximated by Eq. (2.27) using covariance transformation [56]:

$$\int_{\mathbb{R}^n} \boldsymbol{g}(\mathbf{x}) N(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P}) d\mathbf{x} = \int_{\mathbb{R}^n} \boldsymbol{g}(\mathbf{S}\mathbf{x} + \hat{\mathbf{x}}) N(\mathbf{x}; \mathbf{0}, \mathbf{I}) d\mathbf{x} \approx \sum_{i=1}^{N_p} W_i \boldsymbol{g}(\mathbf{S}\boldsymbol{\gamma}_i + \hat{\mathbf{x}}) \tag{2.28}$$

where $\mathbf{P} = \mathbf{S}\mathbf{S}^T$.

Substituting Eq. (2.28) into Eqs. (2.19)-(2.20) and (2.24)-(2.26), the generic point-based Gaussian approximation filter can be obtained as follows.

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{i=1}^{N_p} W_i f(\boldsymbol{\xi}_i) \tag{2.29}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=1}^{N_p} W_i \left(\boldsymbol{f}(\boldsymbol{\xi}_i) - \hat{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{f}(\boldsymbol{\xi}_i) - \hat{\mathbf{x}}_{k|k-1}\right)^T + \mathbf{Q}_{k-1} \tag{2.30}$$

where N_p is the total number of points and $\boldsymbol{\xi}_i$ is the transformed point obtained from the covariance decomposition, i.e.

$$\mathbf{P}_{k-1|k-1} = \mathbf{S}\mathbf{S}^T; \qquad \boldsymbol{\xi}_i = \mathbf{S}\boldsymbol{\gamma}_i + \hat{\mathbf{x}}_{k-1|k-1} \tag{2.31}$$

Note, $\boldsymbol{\gamma}_i$ is the point corresponding to $N(\mathbf{x};\mathbf{0},\mathbf{I}_n)$ and n is the dimension of the state; W_i is the point weight.

Equations (2.21) and (2.23) can still be used and Eqs. (2.24)-(2.26) can be approximated by

$$\hat{\mathbf{y}}_k = \sum_{i=1}^{N_p} W_i \boldsymbol{h}\left(\tilde{\boldsymbol{\xi}}_i\right) \tag{2.32}$$

$$\mathbf{P}_{xy} = \sum_{i=1}^{N_p} W_i \left(\tilde{\boldsymbol{\xi}}_i - \hat{\mathbf{x}}_{k|k-1}\right)\left(\boldsymbol{h}\left(\tilde{\boldsymbol{\xi}}_i\right) - \hat{\mathbf{y}}_k\right)^T \tag{2.33}$$

$$\mathbf{P}_{yy} = \sum_{i=1}^{N_p} W_i \left(\boldsymbol{h}\left(\tilde{\boldsymbol{\xi}}_i\right) - \hat{\mathbf{y}}_k\right)\left(\boldsymbol{h}\left(\tilde{\boldsymbol{\xi}}_i\right) - \hat{\mathbf{y}}_k\right)^T \tag{2.34}$$

where $\tilde{\boldsymbol{\xi}}_i$ is the transformed point obtained from the predicted covariance decomposition, i.e.

$$\mathbf{P}_{k|k-1} = \tilde{\mathbf{S}}\tilde{\mathbf{S}}^T; \quad \tilde{\boldsymbol{\xi}}_i = \tilde{\mathbf{S}}\boldsymbol{\gamma}_i + \hat{\mathbf{x}}_{k|k-1} \tag{2.35}$$

Note that in this framework the points are generated twice within a filter cycle: at the beginning of the filter cycle and immediately after prediction. The points are re-generated after prediction in order to account for the effect of the process noise. An alternative is to generate a larger set of augmented points, each consisting of the state and process noise, only at the beginning of every filter cycle, by applying a point selection algorithm to the augmented covariance matrix consisting of both the state covariance and the process noise covariance [36].

The augmented point-based Gaussian approximation filter can be obtained as follows.

We define the augmented state as

$$\mathbf{x}^a = \begin{bmatrix} \mathbf{x}^T & \mathbf{v}^T & \mathbf{n}^T \end{bmatrix}^T \tag{2.36}$$

and the augmented covariance as

$$\mathbf{P}^a = \begin{bmatrix} \mathbf{P}_x & \mathbf{0}_{n\times n_v} & \mathbf{0}_{n\times n_n} \\ \mathbf{0}_{n_v\times n} & \mathbf{Q} & \mathbf{0}_{n_v\times n_n} \\ \mathbf{0}_{n_n\times n} & \mathbf{0}_{n_n\times n_v} & \mathbf{R} \end{bmatrix} \tag{2.37}$$

where n , n_v , and n_n are the dimension of the state, the dimension of the process noise, and the dimension of the measurement noise, respectively.

Then the prediction step and update step can be rewritten as:

$$\hat{\mathbf{x}}^a_{k|k-1} = \sum_{i=1}^{2(n+n_v+n_n)+1} W_i^a \boldsymbol{f}\left(\boldsymbol{\xi}_i^a\right) \tag{2.38}$$

$$\mathbf{P}^a_{k|k-1} = \sum_{i=1}^{2(n+n_v+n_n)+1} W_i^a \left(\boldsymbol{f}\left(\boldsymbol{\xi}_i^a\right) - \hat{\mathbf{x}}^a_{k|k-1}\right)\left(\boldsymbol{f}\left(\boldsymbol{\xi}_i^a\right) - \hat{\mathbf{x}}^a_{k|k-1}\right)^T \tag{2.39}$$

where $\boldsymbol{\xi}_i^a$ is given by

$$\boldsymbol{\xi}_i^a = \left(\mathbf{S}^a_{k-1|k-1}\right)\boldsymbol{\gamma}_i^a + \hat{\mathbf{x}}^a_{k-1|k-1} \tag{2.40}$$

with

$$\mathbf{P}^a_{k-1|k-1} = \mathbf{S}^a_{k-1|k-1}\left(\mathbf{S}^a_{k-1|k-1}\right)^T \tag{2.41}$$

Note, $\boldsymbol{\gamma}_i^a$ and W_i^a can be obtained by Eq. (2.27).

Similarly, Eqs. (2.21) and Eq.(2.22) can still be used, and Eqs. (2.23)-(2.26) can be rewritten by

$$\mathbf{L}_k = \mathbf{P}_{xy}\left(\mathbf{P}_{yy}\right)^{-1} \tag{2.42}$$

$$\mathbf{P}^a_{xy} = \sum_{i=1}^{2(n+n_v+n_n)+1} W_i^a \left(\boldsymbol{\xi}_i^a - \hat{\mathbf{x}}^a_{k|k-1}\right)\left(\boldsymbol{h}\left(\boldsymbol{\xi}_i^a\right) - \hat{\mathbf{y}}^a_k\right)^T \tag{2.43}$$

$$\mathbf{P}^a_{yy} = \sum_{i=1}^{2(n+n_v+n_n)+1} W_i^a \left(\boldsymbol{h}\left(\boldsymbol{\xi}_i^a\right) - \hat{\mathbf{y}}^a_k\right)\left(\boldsymbol{h}\left(\boldsymbol{\xi}_i^a\right) - \hat{\mathbf{y}}^a_k\right)^T \tag{2.44}$$

$$\hat{\mathbf{y}}^a_k = \sum_{i=1}^{2(n+n_v+n_n)+1} W_i^a \boldsymbol{h}\left(\boldsymbol{\xi}_i^a\right) \tag{2.45}$$

There are many numerical rules that can be used to approximate the integral in Eq. (2.27). They are summarized in References [38-40, 58, 59]. In the following, many typical numerical integral techniques are introduced.

2.2.1.3 Unscented Transformation

The unscented transformation (UT) is an approach to choose points that can approximate the mean and covariance more accurately. By the unscented transformation, the points $\boldsymbol{\gamma}_i$ and weights W_i in Eq. (2.27) are given by

$$\begin{cases} \boldsymbol{\gamma}_1 = \mathbf{0} & W_1 = \kappa/(n+\kappa) & i=1 \\ \boldsymbol{\gamma}_i = \sqrt{(n+\kappa)}\mathbf{e}_{i-1} & W_i = 1/(2(n+\kappa)) & i=2,\cdots,n+1 \\ \boldsymbol{\gamma}_i = -\sqrt{(n+\kappa)}\mathbf{e}_{i-n-1} & W_i = 1/(2(n+\kappa)) & i=n+2,\cdots,2n+1 \end{cases} \tag{2.46}$$

where κ is a parameter to tune the value of $\boldsymbol{\gamma}_i$ and W_i are the weights corresponding to $\boldsymbol{\gamma}_i$.

Take a nonlinear function $\mathbf{y} = \boldsymbol{f}(\mathbf{x})$ for an example, where $\mathbf{y} \in \mathbb{R}^m$; $\mathbf{x} \in \mathbb{R}^n$. To calculate the mean and covariance of $\mathbf{y}$, by Eq.(2.28), $2n+1$ points with different weights are chosen as follows:

$$\begin{cases} \hat{\boldsymbol{\gamma}}_1 = \hat{\mathbf{x}} & W_1 = \kappa/(n+\kappa) & i=1 \\ \hat{\boldsymbol{\gamma}}_i = \hat{\mathbf{x}} + \left(\sqrt{(n+\kappa)\mathbf{P}_{xx}}\right)_i & W_i = 1/(2(n+\kappa)) & i=2,\cdots,n+1 \\ \hat{\boldsymbol{\gamma}}_i = \hat{\mathbf{x}} - \left(\sqrt{(n+\kappa)\mathbf{P}_{xx}}\right)_i & W_i = 1/(2(n+\kappa)) & i=n+2,\cdots,2n+1 \end{cases} \tag{2.47}$$

where $\mathbf{P}_{xx}$ is the covariance of $\mathbf{x}$; $\left(\sqrt{(n+\kappa)\mathbf{P}_{xx}}\right)_i$ is the i^{th} row or column of the matrix $\left(\sqrt{(n+\kappa)\mathbf{P}_{xx}}\right)$ and $(n+\kappa)\mathbf{P}_{xx} = \sqrt{(n+\kappa)\mathbf{P}_{xx}}\left(\sqrt{(n+\kappa)\mathbf{P}_{xx}}\right)^T$. Then the mean and covariance of $\mathbf{y}$ can be approximated by the following equations

$$\hat{\mathbf{y}} \approx \sum_{i=1}^{2n+1} W_i \boldsymbol{f}(\hat{\boldsymbol{\gamma}}_i) \tag{2.48}$$

$$\mathbf{P}_{yy} \approx \sum_{i=1}^{2n+1} W_i \left(\boldsymbol{f}(\hat{\boldsymbol{\gamma}}_i) - \hat{\mathbf{y}}\right)\left(\boldsymbol{f}(\hat{\boldsymbol{\gamma}}_i) - \hat{\mathbf{y}}\right)^T \tag{2.49}$$

It has been proven that the unscented transformation can capture the third order information of the system for a Gaussian system [36].

Different point selection strategies related with the unscented transformation include the symmetric point [35], minimal-skew simplex points [60], spherical simplex

points [61, 62], the Schmidt orthogonalization-based simplex set [63], and those with more than $2n+1$ sigma points for accuracy enhancement [64-66], where n is the dimension. Various sampling methods of the UKF were compared in [67, 68]. For n-dimensional systems ($n>1$), the original UKF using $2n+1$ symmetric points is accurate for the 3rd degree polynomials. With reduced computational cost, the simplex UKF and the spherical simplex UKF can be partly accurate for 3rd degree polynomials using $n+2$ nonsymmetrical points.

2.2.1.4 Central Difference

Let us define $\tilde{\boldsymbol{f}}(\mathbf{x}) = \boldsymbol{f}\left(\mathbf{S}^T\mathbf{x} + \hat{\mathbf{x}}_{k-1|k-1}\right)$, which can be approximated by [56]

$$\tilde{\boldsymbol{f}}(\mathbf{x}) = \tilde{\boldsymbol{f}}(\mathbf{0}) + \sum_{i=1}^{n} a_i x_i + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} \tag{2.50}$$

where x_i is the i^{th} entry of $\mathbf{x}$; a_i, the element $\mathbf{H}_{i,i}$ and $\mathbf{H}_{i,j}$ of the symmetric matrix $\mathbf{H}$ are given by [56]

$$a_i = \frac{\tilde{\boldsymbol{f}}(h\mathbf{e}_i) - \tilde{\boldsymbol{f}}(-h\mathbf{e}_i)}{2h} \qquad 1 \le i \le n \tag{2.51}$$

$$\mathbf{H}_{i,i} = \frac{\tilde{\boldsymbol{f}}(h\mathbf{e}_i) - 2\tilde{\boldsymbol{f}}(0) + \tilde{\boldsymbol{f}}(-h\mathbf{e}_i)}{h^2} \qquad 1 \le i \le n \tag{2.52}$$

$$\mathbf{H}_{i,j} = \frac{\tilde{\boldsymbol{f}}(h\mathbf{e}_i + h\mathbf{e}_j) + \tilde{\boldsymbol{f}}(0) - \tilde{\boldsymbol{f}}(-h\mathbf{e}_i) - \tilde{\boldsymbol{f}}(-h\mathbf{e}_j)}{h^2} \qquad 1 \le i \le j \le n \tag{2.53}$$

where h is a tunable parameter and $\mathbf{e}_j$ is the unit vector in $\mathbb{R}^n$ with the j^{th} element being 1. Then,

$$I_c = \int_{\mathbb{R}^n} \tilde{\boldsymbol{f}}(\mathbf{x}) \frac{1}{(2\pi)^{n/2}} \exp\left(-\mathbf{x}^2/2\right) d\mathbf{x} = \tilde{\boldsymbol{f}}(\mathbf{0}) + \sum_{i=1}^{n} \frac{1}{2}\mathbf{H}_{i,i} \tag{2.54}$$

$$\begin{aligned} &\int_{\mathbb{R}^n} \tilde{\boldsymbol{f}}_1(\mathbf{x})\,\tilde{\boldsymbol{f}}_2(\mathbf{x}) \frac{1}{(2\pi)^{n/2}} \exp\left(-\mathbf{x}^2/2\right) d\mathbf{x} \\ &\quad = I_c^1 I_c^2 + \sum_{i=1}^{n}\left(a_i^1 a_i^2 + \frac{1}{2}\mathbf{H}_{i,i}^1\mathbf{H}_{i,i}^2\right) + \sum_{i\ne j}\mathbf{H}_{i,i}^1\mathbf{H}_{j,j}^2 \end{aligned} \tag{2.55}$$

where $I_c^j = \tilde{\boldsymbol{f}}_j(\mathbf{0}) + \sum_{i=1}^{n} \frac{1}{2} \mathbf{H}_{i,i}^j$ ($j = 1,2$); a_i^j and $\mathbf{H}_{i,i}^j$ related with $\tilde{\boldsymbol{f}}_j(\mathbf{s})$ can be obtained by Eqs. (2.51) and (2.52).

Using Eqs. (2.54) and (2.55), the prediction step can be rewritten as

$$\hat{\mathbf{x}}_{k|k-1} = \tilde{\boldsymbol{f}}(0) + \sum_{i=i}^{n} \frac{1}{2} \mathbf{H}_{i,i} \tag{2.56}$$

$$\mathbf{P}_{k|k-1} = \sum_{i=i}^{n} a_i a_i^T + \sum_{i=i}^{n} \frac{1}{2} \mathbf{H}_{i,i} \mathbf{H}_{i,i}^T + \mathbf{Q} \tag{2.57}$$

Similarly, we define $\tilde{\boldsymbol{h}}(\mathbf{x}) = \boldsymbol{h}\left(\tilde{\mathbf{S}}^T \mathbf{x} + \hat{\mathbf{x}}_{k|k-1}\right)$. Then, it can be approximated by $\tilde{\boldsymbol{h}}(\mathbf{0}) + \sum_{i=1}^{n} b_i x_i + \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x}$. Note, b_i and $\mathbf{G}$ are similar to a_i and $\mathbf{H}$ in Eq. (2.50).

The following equations can be obtained in a similar fashion as the Eqs. (2.56) and (2.57) [56].

$$\mathbf{P}_{xy} = \tilde{\mathbf{S}}^T \left[b_1, \cdots, b_n\right]^T \tag{2.58}$$

$$\mathbf{P}_{yy} = \sum_{i=1}^{n} b_i b_i^T + \sum_{i=1}^{n} \frac{1}{2} \mathbf{G}_{i,i} \mathbf{G}_{i,i}^T \tag{2.59}$$

$$\hat{\mathbf{y}}_k = \tilde{\boldsymbol{h}}(0) + \sum_{i=1}^{n} \frac{1}{2} \mathbf{G}_{i,i} \tag{2.60}$$

More details can be found in [56].

2.2.1.5 Gauss-Hermite Quadrature

For the univariate Gauss-Hermite quadrature (GHQ) rule that represents the univariate standard Gaussian distribution $N(x;0,1)$ with m quadrature points, γ_i and w_i can be calculated as follows [29, 56]. If $m = 1$, then $\gamma_1 = 0$ and $w_1 = 1$. If $m > 1$, first construct a symmetric tri-diagonal matrix J with zero diagonal elements and $J_{i,i+1} = J_{i+1,i} = \sqrt{i/2}$, $1 \le i \le m-1$. Then the quadrature point γ_i is calculated by $\gamma_i = \sqrt{2}\varepsilon_i$, where ε_i is the i^{th} eigenvalue of J. The corresponding w_i is calculated by

$w_i = (\mathbf{v}_i)_1^2$ where $(\mathbf{v}_i)_1$ is the first element of the i^{th} normalized eigenvector of J. The univariate GHQ rule with m points is exact up to $(2m-1)^{\text{th}}$ order polynomials [56].

To represent $N(\mathbf{x};\mathbf{0},\mathbf{I}_n)$, the multivariate GHQ rule extends the univariate m-point set to the n-dimensional point set by the tensor product rule [29, 56]. It is exact for all polynomials of the form $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ with $1 \le i_j \le 2m-1$ [29, 56]. However, the total number of points $N_p = m^n$ increases exponentially with the dimension n.

For example, the two dimensional GHQ points $\boldsymbol{\gamma}_i$ and weights W_i constructed from 3 univariate GHQ points $\gamma_1, \gamma_2, \gamma_3$ with corresponding weights w_1, w_2, w_3 by the tensor product rule contains 3^2=9 points

$$\begin{pmatrix} (\gamma_1,\gamma_1),(\gamma_1,\gamma_2),(\gamma_1,\gamma_3) \\ (\gamma_2,\gamma_1),(\gamma_2,\gamma_2),(\gamma_2,\gamma_3) \\ (\gamma_3,\gamma_1),(\gamma_3,\gamma_2),(\gamma_3,\gamma_3) \end{pmatrix}$$

and 9 corresponding weights

$$\begin{pmatrix} w_1 w_1, w_1 w_2, w_1 w_3 \\ w_2 w_1, w_2 w_2, w_2 w_3 \\ w_3 w_1, w_3 w_2, w_3 w_3 \end{pmatrix}$$

where $\gamma_1 = -\sqrt{3}$, $\gamma_2 = 0$, and $\gamma_3 = \sqrt{3}$; $w_1 = 1/6$, $w_2 = 2/3$, and $w_3 = 1/6$.

2.2.1.6 Cubature Rules

Cubature rules, also called spherical-radial rules, have been used in multiple integrals [37, 69, 70]. Cubature Kalman filters with different accuracies can be obtained by using cubature rules with different degrees of accuracy in the framework of point-based Gaussian approximation filters. Recently, the third-degree cubature rule was used to solve the nonlinear estimation problem in [37].

Before introducing the cubature rules, we consider the following integral.

$$I(\boldsymbol{g}) = \int_{\mathbb{R}^n} \boldsymbol{g}(\mathbf{x}) \exp\left(-\mathbf{x}^T\mathbf{x}\right) d\mathbf{x} = \int_0^\infty \int_{U_n} \boldsymbol{g}(r\mathbf{s}) r^{n-1} \exp\left(-r^2\right) d\sigma(\mathbf{s}) dr \tag{2.61}$$

where $\mathbf{s} = [s_1, s_2, \cdots, s_n]^T$, $U_n = \left\{\mathbf{s} \in \mathbb{R}^n : s_1^2 + s_2^2 + \cdots + s_n^2 = 1\right\}$, and $\sigma(\cdot)$ is the spherical measure on U_n [37].

Equation (2.61) contains two different types of integrals. The first type of integral $\int_0^\infty g(r) r^{n-1} \exp\left(-r^2\right) dr$ is the radial integral [37, 69, 70] and the second type of integral $\int_{U_n} \boldsymbol{g}(\mathbf{s}) d\sigma(\mathbf{s})$ is the spherical integral [37, 69, 70]. The cubature rule is based on the combination of these two types of integrals and the following fact [37, 69, 70]: If the radial integral can be approximated by the N_r-point radial rule

$$\int_0^\infty g(r) r^{n-1} \exp\left(-r^2\right) dr \approx \sum_{i=1}^{N_r} w_{r,i} g(r_i) \tag{2.62}$$

and the spherical integral can be approximated by N_s-point spherical rule,

$$\int_{U_n} \boldsymbol{g}(\mathbf{s}) d\sigma(\mathbf{s}) = \sum_{j=1}^{N_s} w_{s,j} \boldsymbol{g}(\mathbf{s}_j) \tag{2.63}$$

where r_i and $w_{r,i}$ are points and weights for calculating the radial integral, $\mathbf{s}_j$ and $w_{s,j}$ are points and weights for calculating the spherical integral, then Eq. (2.61) can be computed by

$$\begin{aligned} I(\boldsymbol{g}) &= \int_0^\infty r^{n-1} \exp\left(-r^2\right) \int_{U_n} \boldsymbol{g}(r\mathbf{s}) d\sigma(\mathbf{s}) dr \approx \int_0^\infty r^{n-1} \exp\left(-r^2\right) \sum_{j=1}^{N_s} w_{s,j} \boldsymbol{g}(r\mathbf{s}_j) dr \\ &\approx \sum_{i=1}^{N_r} \sum_{j=1}^{N_s} w_{r,i} w_{s,j} \boldsymbol{g}(r_i \mathbf{s}_j) \end{aligned} \tag{2.64}$$

The third-degree spherical rule is given by [37]

$$I_{U_n,3}(\boldsymbol{g}) = \frac{A_n}{2n} \sum_{j=1}^{n} \left(\boldsymbol{g}(\mathbf{e}_j) + \boldsymbol{g}(-\mathbf{e}_j)\right) \tag{2.65}$$

where $A_n = 2\Gamma(1/2)^n / \Gamma(n/2) = 2\sqrt{\pi^n} / \Gamma(n/2)$ is the surface area of the unit sphere; $\Gamma(n)$ is the gamma function defined by the integral $\Gamma(n) = \int_0^\infty \exp(-\lambda)\lambda^{n-1} d\lambda$; $\mathbf{e}_j$ is the unit vector in $\mathbb{R}^n$ with the j^{th} element being 1.

The third-degree radial rules can be obtained by the moment matching method. The general moment matching method can be described by

$$\int_0^\infty g(r) r^{n-1} \exp(-r^2) dr = \sum_{i=1}^{N_r} w_{r,i} g(r_i) \tag{2.66}$$

where $g(r)$ is a polynomial.

For the third-degree radial rule, the following equations can be obtained.

$$\begin{cases} w_{r,1} r_1^0 = \dfrac{1}{2}\Gamma\left(\dfrac{1}{2}n\right) \\ w_{r,1} r_1^2 = \dfrac{1}{2}\Gamma\left(\dfrac{1}{2}n+1\right) = \dfrac{n}{4}\Gamma\left(\dfrac{1}{2}n\right) \end{cases} \tag{2.67}$$

Note, the terms on the right-hand side of Eq. (2.67) are the exact moments in Eq. (2.66).

Solving Eq. (2.67), the point and weight for the third-degree radial rule can be obtained.

$$r_1 = \sqrt{\frac{n}{2}} \tag{2.68}$$

$$w_{r,1} = \frac{\Gamma(n/2)}{2} \tag{2.69}$$

Hence, the third-degree cubature rule is given by

$$\begin{aligned} &\int_{\mathbb{R}^n} \boldsymbol{g}(\mathbf{x}) N(\mathbf{x}; \mathbf{0}, \mathbf{I}) d\mathbf{x} = \frac{1}{\pi^{n/2}} \int_{\mathbb{R}^n} \boldsymbol{g}\left(\sqrt{2}\mathbf{x}\right) \exp\left(-\mathbf{x}^T\mathbf{x}\right) d\mathbf{x} \\ &= \frac{1}{\pi^{n/2}} \int_0^\infty \int_{U_n} \boldsymbol{g}\left(\sqrt{2} r\mathbf{s}\right) r^{n-1} \exp\left(-r^2\right) d\sigma(\mathbf{s}) dr \\ &\approx \frac{1}{\pi^{n/2}} \sum_{i=1}^{N_r} \sum_{j=1}^{N_s} w_{r,i} w_{s,j} \boldsymbol{g}\left(\sqrt{2} r_i \mathbf{s}_j\right) \\ &= \frac{1}{\pi^{n/2}} \sum_{j=1}^{n} \frac{\Gamma(n/2)}{2} \frac{A_n}{2n} \left(\boldsymbol{g}\left(\sqrt{2}\sqrt{\frac{n}{2}} \times \mathbf{e}_j \right) + \boldsymbol{g}\left(-\sqrt{2}\sqrt{\frac{n}{2}} \times \mathbf{e}_j \right) \right) \\ &= \frac{1}{2n} \sum_{j=1}^{n} \left(\boldsymbol{g}\left(\sqrt{n} \times \mathbf{e}_j \right) + \boldsymbol{g}\left(-\sqrt{n} \times \mathbf{e}_j \right) \right) \end{aligned} \tag{2.70}$$

The higher-degree cubature rules can be obtained by combining the higher-degree spherical rules and the higher-degree radial rules.

2.2.1.7 Divided Difference Filters

The divided difference filters are based on Stirling"s interpolation. In this section, the first order and the second order Stirling"s interpolation are introduced.

2.2.1.7.1 The First Order Stirling's Interpolation

The first order Stirling"s interpolation can be described by [57]

$$\mathbf{y} = \boldsymbol{f}\left(\mathbf{x}\right) = \bar{\boldsymbol{f}}\left(\tilde{\boldsymbol{\delta}}\right) \approx \bar{\boldsymbol{f}}\left(\hat{\boldsymbol{\delta}}\right) + \sum_{i=1}^{n} \Delta\tilde{\zeta}_i \frac{\bar{\boldsymbol{f}}\left(\hat{\boldsymbol{\delta}} + h\mathbf{e}_i\right) - \bar{\boldsymbol{f}}\left(\hat{\boldsymbol{\delta}} - h\mathbf{e}_i\right)}{2h} \tag{2.71}$$

where $\tilde{\boldsymbol{\delta}} = \mathbf{S}^{-1}\mathbf{x}$; $\mathbf{S}$ is calculated from covariance factorization of covraince $\mathbf{P}$, i. e. $\mathbf{P} = \mathbf{S}\mathbf{S}^T$; $\Delta\tilde{\zeta}_i$ is the i^{th} element of vector $\tilde{\boldsymbol{\delta}} - \hat{\boldsymbol{\delta}}$ ($\hat{\boldsymbol{\delta}}$ is a reference value), and h is the interpolation interval.

The mean and covariance can be calculated by the first order Stirling"s interpolation [57, 71]

$$\hat{\mathbf{y}}^{DD1} = \boldsymbol{f}\left(\hat{\mathbf{x}}\right) \tag{2.72}$$

$$\mathbf{P}_{yy}^{DD1} = \frac{1}{4h^2}\sum_{i=1}^{n}\left(\boldsymbol{f}\left(\hat{\mathbf{x}} + h\mathbf{S}_i\right) - \boldsymbol{f}\left(\hat{\mathbf{x}} - h\mathbf{S}_i\right)\right)\left(\boldsymbol{f}\left(\hat{\mathbf{x}} + h\mathbf{S}_i\right) - \boldsymbol{f}\left(\hat{\mathbf{x}} - h\mathbf{S}_i\right)\right)^T \tag{2.73}$$

$$\mathbf{P}_{xy}^{DD1} = \frac{1}{2h}\sum_{i=1}^{n}\mathbf{S}_i\left(\boldsymbol{f}\left(\hat{\mathbf{x}} + h\mathbf{S}_i\right) - \boldsymbol{f}\left(\hat{\mathbf{x}} - h\mathbf{S}_i\right)\right)^T \tag{2.74}$$

where $\mathbf{S}_i$ is the i^{th} column of the matrix $\mathbf{S}$.

2.2.1.7.2 The Second Order Stirling's Interpolation

The second order Stirling"s interpolation can be described by

$$\begin{aligned}\mathbf{y} = \boldsymbol{f}(\mathbf{x}) = \bar{\boldsymbol{f}}(\tilde{\boldsymbol{\delta}}) \approx \bar{\boldsymbol{f}}(\hat{\boldsymbol{\delta}}) + \sum_{i=1}^{n} \Delta\tilde{\zeta}_i \frac{\bar{\boldsymbol{f}}(\hat{\boldsymbol{\delta}} + h\mathbf{e}_i) - \bar{\boldsymbol{f}}(\hat{\boldsymbol{\delta}} - h\mathbf{e}_i)}{2h} \\ + \sum_{i=1}^{n} (\Delta\tilde{\zeta}_i)^2 \frac{\bar{\boldsymbol{f}}(\hat{\boldsymbol{\delta}} + h\mathbf{e}_i) + \bar{\boldsymbol{f}}(\hat{\boldsymbol{\delta}} - h\mathbf{e}_i) - 2\bar{\boldsymbol{f}}(\hat{\boldsymbol{\delta}})}{2h^2}\end{aligned} \tag{2.75}$$

The mean and covariance can be calculated by the second order Stirling"s interpolation [57, 71]

$$\hat{\mathbf{y}}^{DD2} = \frac{h^2 - n_x}{h^2} \boldsymbol{f}(\hat{\mathbf{x}}) + \frac{1}{2h^2} \sum_{i=1}^{n} \left(\boldsymbol{f}(\hat{\mathbf{x}} + h\mathbf{S}_i) + \boldsymbol{f}(\hat{\mathbf{x}} - h\mathbf{S}_i) \right) \tag{2.76}$$

$$\begin{aligned}\mathbf{P}_{yy}^{DD2} = \mathbf{P}_{yy}^{DD1} + \frac{h^2 - 1}{4h^4} \sum_{i=1}^{n} \left(\boldsymbol{f}(\hat{\mathbf{x}} + h\mathbf{S}_i) + \boldsymbol{f}(\hat{\mathbf{x}} - h\mathbf{S}_i) - 2\boldsymbol{f}(\hat{\mathbf{x}}) \right) \\ \times \left(\boldsymbol{f}(\hat{\mathbf{x}} + h\mathbf{S}_i) + \boldsymbol{f}(\hat{\mathbf{x}} - h\mathbf{S}_i) - 2\boldsymbol{f}(\hat{\mathbf{x}}) \right)^T\end{aligned} \tag{2.77}$$

$$\mathbf{P}_{xy}^{DD2} = \frac{1}{2h} \sum_{i=1}^{n} \mathbf{S}_i \left(\boldsymbol{f}(\hat{\mathbf{x}} + h\mathbf{S}_i) - \boldsymbol{f}(\hat{\mathbf{x}} - h\mathbf{S}_i) \right)^T \tag{2.78}$$

For the Gaussian distribution, the optimal choice of the interval is $h^2 = 3$.

Note that the mean computation of the unscented transformation and the second order Stirling"s interpolation are the same [57, 71]. In addition, the cross covariance matrices calculated by the unscented transformation and the Stirling"s interpolation are the same when $n + \kappa = h^2$ [57].

2.2.1.8 Discussions

Compared with the multivariate GHQ rule, the unscented transformation (UT) and the cubature rule are much more efficient computationally because the number of points used or the computational cost is linear with respect to the dimension of the state. However, the accuracy that can be achieved by the UT and the cubature rule is much lower than the GHQ rule. Take the UT with 2*n*+1 sigma points as an example. Without

loss of generality, assume the covariance of the state is an $n \times n$ identity matrix and the UT uses three distinct sigma points for each dimension. The UT can calculate the expectation of $x_i^k, i = 1, \cdots, n, 0 \le k \le 5$ exactly. But the high accuracy of the UT for univariate polynomials does not imply that the UT is exact for multivariate polynomials up to the 5th order. It is a known fact that the standard UT fails to calculate the expectation of $x_i^2 x_j^2$, $(i, j = 1, \cdots, n \text{ and } i \ne j)$ exactly [36]. The sparse Gauss-Hermite quadrature, which will be introduced in the next chapter, may be considered as a more accurate UT. It can achieve guaranteed levels of accuracy for multivariate problems when calculating moments (expectations of polynomials of state variables).

2.2.2 Non-Gaussian Approximation Filters

There are mainly three types of nonlinear filtering methods in this category: the sequential Monte-Carlo filter [3, 6, 7, 32, 43, 44, 46, 72-75], the point-mass filter [27, 28, 76], and the Gaussian sum filter [3, 33, 45, 77-80]. The sequential Monte-Carlo method is based on simulations using a set of weighted particles drawn from a proposal distribution to simulate the dynamic process and measurement process [32, 81]. The pdf can be approximated by those particles. This method is easy to implement. However, it is computationally intensive and the direct implementation leads to degeneration problem [32, 81]. The computational problem is alleviated with the advancement of computation capabilities in recent years. The degeneration problem is that, in practice, after a few iterations of the algorithm, all but one of the normalized weights are very close to zero and large computational effort is devoted to updating particles whose contribution to the final estimate is almost zero. To solve this problem, many new improvements have been proposed [6, 32, 81]. Resampling is the most frequently used method. A suitable metric

of degeneracy of the algorithm was introduced. When this value is greater than the expected value, a resampling strategy is used. Because the sampling step is difficult to implement for non-Gaussian distributions, proposal sampling distribution is used to provide sampling particles [6, 32, 81]. The EKF, UKF and GHQF have been used to obtain a good proposal distribution [34, 82, 83]. Although the computational efficiency has been improved in recent decades, the large computational burden for high dimensional problems still exists, since a huge number of particles is needed [43]. Different from the point-selection strategy of the sequential Monte-Carlo method, the point-mass filter uses a set of deterministic points associated with specific masses to represent the pdf. Although it works well for lower dimensional problems, the point-mass filter is difficult to use for high dimensional problems, since the computational burden is large. The Gaussian sum filter is based on the theorem that any probability density function can be approximated by the sum of finite Gaussian distributions. The non-Gaussian distribution is first described by finite Gaussian distributions. Then, each Gaussian component uses the procedure of the Gaussian approximation filter, such as EKF, UKF, or GHQF.

2.2.2.1 Sequential Monte Carlo Method

The integrals in the Chapman-Kolmogorov equation and the Bayesian equation can be calculated by the Monte Carlo method. Assume we are able to simulate N_s independent and identically distributed random particles $\left\{\mathbf{x}_{1:k}^{(i)}; i=1,\cdots,N_s\right\}$ according to the pdf $p\left(\mathbf{x}_{1:k} \mid \mathbf{y}_{1:k}\right)$. An approximation of this distribution is given by

$$p_{N_s}\left(\mathbf{x}_{1:k} \mid \mathbf{y}_{1:k}\right) = \frac{1}{N_s}\sum_{i=1}^{N_s}\delta\left(\mathbf{x}_{1:k} - \mathbf{x}_{1:k}^{(i)}\right) \tag{2.79}$$

where $\delta(\cdot)$ denotes the Dirac delta function.

Hence, the integral $I(\boldsymbol{f}) = \int \boldsymbol{f}(\mathbf{x}_{1:k})\, p(\mathbf{x}_{1:k} \mid \mathbf{y}_{1:k})\, d\mathbf{x}_{1:k}$ can be approximated by

$$I_{N_s}(\boldsymbol{f}) = \int \boldsymbol{f}(\mathbf{x}_{1:k})\, p_{N_s}(\mathbf{x}_{1:k} \mid \mathbf{y}_{1:k})\, d\mathbf{x}_{1:k} = \frac{1}{N_s} \sum_{i=1}^{N_s} \boldsymbol{f}\left(\mathbf{x}_{1:k}^{(i)}\right) \tag{2.80}$$

From the set of random particles $\left\{\mathbf{x}_{1:k}^{(i)}; i = 1, \cdots, N_s\right\}$, we can easily estimate the value of $I(\boldsymbol{f})$. When the number of particles N_s goes to infinity, $I_{N_s}(\boldsymbol{f})$ will be equal to the true $I(\boldsymbol{f})$.

The merit of the sequential Monte Carlo method is that the central limit theorem guarantees that this type of method has a convergence rate independent of the dimension of the integrand. In contrast, any deterministic numerical integration method has a rate of convergence that decreases with increasing dimension [32].

For a general non-Gaussian, multivariate or multi-modal pdf $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$, it may be difficult to generate samples from $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$. To overcome this difficulty, we assume that $q(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ is another pdf from which samples can be easily drawn. Then we can write $p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \propto q(\mathbf{x}_k \mid \mathbf{y}_{1:k})$, which means $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ is proportional to $q(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ at every $\mathbf{x}_k$. The scaling factor can be written as

$$\omega(\mathbf{x}_k) = \frac{p(\mathbf{x}_k \mid \mathbf{y}_{1:k})}{q(\mathbf{x}_k \mid \mathbf{y}_{1:k})} \tag{2.81}$$

Hence,

$$E\left[\boldsymbol{f}(\mathbf{x}_k)\right]_{p(\mathbf{x}_k \mid \mathbf{y}_{1:k})} = \frac{\int \boldsymbol{f}(\mathbf{x}_k)\, \omega(\mathbf{x}_k)\, q(\mathbf{x}_k \mid \mathbf{y}_{1:k})\, d\mathbf{x}_k}{\int \omega(\mathbf{x}_k)\, q(\mathbf{x}_k \mid \mathbf{y}_{1:k})\, d\mathbf{x}_k} \tag{2.82}$$

where $E[\cdot]_p$ denotes the expected value with pdf p.

If N_s particles $\left\{\mathbf{x}_k^{(i)}, i = 1, \cdots, N_s\right\}$ are generated from $q(\mathbf{x}_k \mid \mathbf{y}_{1:k})$, Eq. (2.82) can be rewritten as

$$E\left[\boldsymbol{f}(\mathbf{x}_k)\right]_{p(\mathbf{x}_k \mid \mathbf{y}_{1:k})} \approx \frac{\frac{1}{N_s} \sum_{i=1}^{N_s} \boldsymbol{f}\left(\mathbf{x}_k^{(i)}\right) \omega\left(\mathbf{x}_k^{(i)}\right)}{\frac{1}{N_s} \sum_{i=1}^{N_s} \omega\left(\mathbf{x}_k^{(i)}\right)} = \sum_{i=1}^{N_s} \boldsymbol{f}\left(\mathbf{x}_k^{(i)}\right) \tilde{\omega}\left(\mathbf{x}_k^{(i)}\right) \tag{2.83}$$

where

$$\tilde{\omega}\left(\mathbf{x}_k^{(i)}\right) = \frac{\omega\left(\mathbf{x}_k^{(i)}\right)}{\frac{1}{N_s}\sum_{i=1}^{N_s}\omega\left(\mathbf{x}_k^{(i)}\right)} \tag{2.84}$$

Suppose we have a set of particles and weights that characterizes a posterior pdf for times up to t_{k-1}. Then a posterior pdf can be approximated by

$$p\left(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}\right) \approx \sum_{i=1}^{N_s} \omega_{k-1}^{(i)} \delta\left(\mathbf{x}_{k-1} - \mathbf{x}_{k-1|k-1}^{(i)}\right) \tag{2.85}$$

and

$$\omega_{k-1}^{(i)} = \frac{p\left(\mathbf{x}_{k-1|k-1}^{(i)}\right)}{q\left(\mathbf{x}_{k-1|k-1}^{(i)}\right)} \tag{2.86}$$

The importance sampling method can be modified to calculate an approximation of the current pdf without modifying past simulated trajectories.

$$q\left(\mathbf{x}_{1:k} \mid \mathbf{y}_{1:k}\right) = q\left(\mathbf{x}_{1:k-1} \mid \mathbf{y}_{1:k-1}\right) q\left(\mathbf{x}_k \mid \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k}\right) \tag{2.87}$$

Eq. (2.87) can be rewritten as follows:

$$q\left(\mathbf{x}_{1:k} \mid \mathbf{y}_{1:k}\right) = q\left(\mathbf{x}_1\right) \prod_{k=1}^{N_s} q\left(\mathbf{x}_k \mid \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k}\right) \tag{2.88}$$

We want to approximate $p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k}\right)$ with the new set of samples and weights. The weight update equation for each particle becomes

$$\omega_k^{(i)} \propto \omega_{k-1}^{(i)} \frac{p\left(\mathbf{y}_k \mid \mathbf{x}_{k|k-1}^{(i)}\right) p\left(\mathbf{x}_{k|k-1}^{(i)} \mid \mathbf{x}_{k-1|k-1}^{(i)}\right)}{q\left(\mathbf{x}_{k|k-1}^{(i)} \mid \mathbf{x}_{k-1|k-1}^{(i)}\right)} \tag{2.89}$$

where $\mathbf{x}_{k|k-1}^{(i)}$ are given by

$$\mathbf{x}_{k|k-1}^{(i)} = \boldsymbol{f}\left(\mathbf{x}_{k-1|k-1}^{(i)}, \mathbf{v}_{k-1}^{(i)}\right) \tag{2.90}$$

with $\mathbf{v}_{k-1}^{(i)}$ being the random sample from the pdf of the process noise.

Finally, the posterior filtered pdf $p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k}\right)$ can be approximated by

$$p\left(\mathbf{x}_k \mid \mathbf{y}_{1:k}\right) \approx \sum_{i=1}^{N_s} \omega_k^{(i)} \delta\left(\mathbf{x}_k - \mathbf{x}_{k|k-1}^{(i)}\right) \tag{2.91}$$

Unfortunately, the unconditional variance of the importance weights increases over time [32], In practice, after a few iterations of the algorithm, all but one of the normalized importance weights are very close to zero and large computational effort is devoted to updating trajectories whose contribution to the final estimate is almost zero. The resampling method eliminates trajectories or samples with small normalized importance weights and concentrates upon trajectories or samples with large weights. A suitable measure of degeneracy of the algorithm is the effective sample size N_{eff} which is defined as

$$N_{eff} = \frac{1}{\sum_{i=1}^{N_s} \left(\omega_k^{(i)}\right)^2} \tag{2.92}$$

If the effective sample size is less than a given threshold N_{thr}, then resampling is performed. There are also other resampling strategies [32].

We use the bootstrap filter as an example to illustrate the procedure of the particle filter. The bootstrap filter uses the state transition density $\boldsymbol{f}$ as the importance sampling distribution. The corresponding importance weights then are simplified to $\omega_k^{(i)} \propto \omega_{k-1}^{(i)} p\left(\mathbf{y}_k \mid \mathbf{x}_{k|k-1}^{(i)}\right)$.

The procedure is given as follows:

Initialization

For $i = 1, \cdots, N_s$, sample $\tilde{\mathbf{x}}_0^{(i)} = p(\mathbf{x}_0)$

Importance Sampling Step

1. For $i = 1, \cdots, N_s$, sample $\tilde{\mathbf{x}}_k^{(i)}$ from $p\left(\mathbf{x}_k \mid \tilde{\mathbf{x}}_{1:k-1}^{(i)}\right)$, i. e. $\tilde{\mathbf{x}}_k^{(i)} = \boldsymbol{f}\left(\mathbf{x}_k \mid \tilde{\mathbf{x}}_{1:k-1}^{(i)}\right)$.
2. For $i = 1, \cdots, N_s$, evaluate the importance weights $\omega_k^{(i)} = \omega_{k-1}^{(i)} p\left(\mathbf{y}_k \mid \tilde{\mathbf{x}}_k^{(i)}\right)$.
3. Normalize the importance weights, $\tilde{\omega}_k^{(i)} = \omega_k^{(i)} \Big/ \sum_{j=1}^{N_s} \omega_k^{(i)}$.

Resampling step

1. Obtain N_s replacement particles $\left(\left[\tilde{\mathbf{x}}_{0:k}^{(i)}\right]_{new}; i=1,\cdots,N_s\right)$ from the old set $\left(\left[\tilde{\mathbf{x}}_{0:k}^{(i)}\right]_{old}; i=1,\cdots,N_s\right)$ according to the normalized importance weights $\tilde{\omega}_k^{(i)}$; let $\omega_k^{(i)}=1/N_s$.
2. Use the N_s replacement particles $\left(\left[\tilde{\mathbf{x}}_{0:k}^{(i)}\right]_{new}; i=1,\cdots,N_s\right)$ and $\omega_k^{(i)}$ in the importance sampling step to propagate particles.

2.2.2.2 Point-Mass Filter

Assume the pdf can be represented by a set of deterministic points $\left\{\mathbf{x}^{(i)}; i=1,\cdots,N_s\right\}$ with a set of volume masses $\left\{\omega^{(i)}; i=1,\cdots,N_s\right\}$ of each point. The probability values of points are $\left\{p_{\mathbf{x}}^{(i)}; i=1,\cdots,N_s\right\}$. Then the prediction step and update step of the Bayesian filtering can be rewritten as

Prediction:

$$p_{\mathbf{x},k|k-1}^{(i)}=\sum_{i=1}^{N_s}\omega_{k-1}^{(i)}p_{\mathbf{x},k-1|k-1}^{(i)}p_{\mathbf{v},k-1}^{(i)}\left(\mathbf{x}_{k|k-1}^{(i)}-\boldsymbol{f}\left(\mathbf{x}_{k-1|k-1}^{(i)}\right)\right) \tag{2.93}$$

where $\mathbf{x}_{k|k-1}^{(i)}$ are predicted points determined by a specific algorithm [27, 28]. $p_{\mathbf{v},k-1}^{(i)}(\cdot)$ is the probability associated with the process noise pdf at a specific point.

Update:

$$p_{\mathbf{x},k|k}^{(i)}=c_k^{-1}p_{\mathbf{x},k|k-1}^{(i)}p_{\mathbf{n},k}^{(i)}\left(\mathbf{x}_{k|k}^{(i)}-\boldsymbol{h}\left(\mathbf{x}_{k|k-1}^{(i)}\right)\right) \tag{2.94}$$

where $c_k=\sum_{i=1}^{N_s}\omega_k^{(i)}p_{\mathbf{x},k|k-1}^{(i)}p_{\mathbf{n},k}^{(i)}\left(\mathbf{x}_{k|k}^{(i)}-\boldsymbol{h}\left(\mathbf{x}_{k|k-1}^{(i)}\right)\right)$; $p_{\mathbf{n},k-1}^{(i)}(\cdot)$ is the probability associated with the measurement noise pdf at a specific point. More details about the point design of $\mathbf{x}_{k|k-1}^{(i)}$ and $\mathbf{x}_{k|k}^{(i)}$ can be found in [27, 28].

The main disadvantages of the point-mass filter are: 1) it is computationally intensive and hard to use for high-dimensional problems because of the large number of points that is required; 2) the point design for $\mathbf{x}_{k|k-1}^{(i)}$ and $\mathbf{x}_{k|k}^{(i)}$ is complicated.

2.2.2.3 Gauss Sum Filter

Non-Gaussian noise distribution can be approximated by a finite sum of Gaussian distributions. Any probability density function $p_{\mathbf{x}}$ can be approximated by [29]

$$p_{\mathbf{x}} \approx \sum_{i=1}^{N_{\mathbf{x}}} a_i N\left(\mathbf{x}; \mathbf{x}_i, \mathbf{P}_i\right) \tag{2.95}$$

where $N_{\mathbf{x}}$ is the number of Gaussian distributions; a_i is the coefficient for the i^{th} Gaussian distribution $N\left(\mathbf{x}; \mathbf{x}_i, \mathbf{P}_i\right)$ and $\sum_{i=1}^{N_{\mathbf{x}}} a_i = 1$.

Based on Eq. (2.95), we assume that the initial pdf $p_{\mathbf{x},0}$, the pdf of the process noise, and the pdf of the measurement noise at time k can be approximated as

$$p_{\mathbf{x},0} = \sum_{i=1}^{N_{\mathbf{x},0}} \alpha_{\mathbf{x},0,i} p_{\mathbf{x},0,i}\left(\mathbf{x}_0\right) = \sum_{i=1}^{N_{\mathbf{x},0}} \alpha_{\mathbf{x},0,i} N\left(\mathbf{x}_0; \hat{\mathbf{x}}_{0,i}, \mathbf{P}_{0,i}\right) \tag{2.96}$$

$$p_{\mathbf{v},k} = \sum_{i=1}^{N_{\mathbf{v}}} \alpha_{\mathbf{v},k,i} N\left(\mathbf{v}_k; \hat{\mathbf{v}}_{k,i}, \mathbf{Q}_{k,i}\right) \tag{2.97}$$

$$p_{\mathbf{n},k} = \sum_{i=1}^{N_{\mathbf{n}}} \alpha_{\mathbf{n},k,i} N\left(\mathbf{n}_k; \hat{\mathbf{n}}_{k,i}, \mathbf{R}_{k,i}\right) \tag{2.98}$$

where coefficients satisfy

$$\sum_{i=1}^{N_{\mathbf{x},0}} \alpha_{\mathbf{x},0,i} = 1 \quad \sum_{i=1}^{N_{\mathbf{v}}} \alpha_{\mathbf{v},k,i} = 1 \quad \sum_{i=1}^{N_{\mathbf{n}}} \alpha_{\mathbf{n},k,i} = 1$$

Because the pdf at time $k-1$ can be approximated by a finite number of Gaussian distributions, each Gaussian distribution component can be predicted separately.

$$p\left(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}\right) = \sum_{i=1}^{N_{\mathbf{x},k-1}} \alpha_{\mathbf{x},k-1,i} N\left(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1,i}, \mathbf{P}_{k-1,i}\right) \tag{2.99}$$

The transition pdf at time k can be approximated by $N_{\mathbf{v}}$ Gaussian distributions as

$$p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) = \sum_{j=1}^{N_{\mathbf{v}}} \alpha_{\mathbf{v},k,j} N\left(\mathbf{x}_k; \boldsymbol{f}(\mathbf{x}_{k-1}) + \hat{\mathbf{v}}_{k,j}, \mathbf{Q}_{k,j}\right) \tag{2.100}$$

The predicted pdf at time k can be described by

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = & \sum_{i=1}^{N_{\mathbf{x},k-1}} \sum_{j=1}^{N_{\mathbf{v},k}} \alpha_{\mathbf{x},k-1,i} \alpha_{\mathbf{v},k,j} \int_{\mathbb{R}^n} N\left(\mathbf{x}; \hat{\mathbf{x}}_{k|k-1,i}, \mathbf{P}_{k|k-1,i}\right) \\ & \times N\left(\mathbf{x}_k; \boldsymbol{f}(\mathbf{x}) + \hat{\mathbf{v}}_{k,j}, \mathbf{Q}_{k,j}\right) d\mathbf{x} \end{aligned} \tag{2.101}$$

It can be rewritten as

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \sum_{i=1}^{N_{\mathbf{x},k-1}} \sum_{j=1}^{N_{\mathbf{v},k}} \alpha_{\mathbf{x},k-1,i} \alpha_{\mathbf{v},k,j} N\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1,i,j}, \mathbf{P}_{k|k-1,i,j}\right) \tag{2.102}$$

where $\hat{\mathbf{x}}_{k|k-1,i,j}$ and $\mathbf{P}_{k|k-1,i,j}$ are calculated by Gaussian approximation filters, such as, EKF and UKF.

Reordering the Gaussian components of $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})$, we have

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \sum_{r=1}^{N_{\mathbf{x},k}} \alpha_{\mathbf{x},k,r} N\left(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1,r}, \mathbf{P}_{k|k-1,r}\right) \tag{2.103}$$

where $N_{\mathbf{x},k} = N_{\mathbf{x},k-1} \cdot N_{\mathbf{v},k}$, $\alpha_{\mathbf{x},k,r} = \alpha_{\mathbf{x},k-1,i} \cdot \alpha_{\mathbf{v},k,j}$, $\hat{\mathbf{x}}_{k|k-1,r} = \hat{\mathbf{x}}_{k|k-1,i,j}$, and $\mathbf{P}_{k|k-1,r} = \mathbf{P}_{k|k-1,i,j}$.

Assume the likelihood function $p(\mathbf{y}_k \mid \mathbf{x}_k)$ at time k can be approximated by[29]

$$p(\mathbf{y}_k \mid \mathbf{x}_k) = \sum_{j=1}^{N_{\mathbf{n}}} \alpha_{\mathbf{n},k,j} N\left(\mathbf{y}_k; \boldsymbol{h}(\mathbf{x}_k) + \hat{\mathbf{n}}_{k,j}, \mathbf{R}_{k,j}\right) \tag{2.104}$$

The updated pdf can be calculated by

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \approx \sum_{r=1}^{N_{\mathbf{x},k} \cdot N_{\mathbf{n},k}} \alpha_{k|k,r} N\left(\hat{\mathbf{x}}_{k|k,r}, \mathbf{P}_{k|k,r}\right) \tag{2.105}$$

where $\alpha_{k|k,r} = \dfrac{\alpha_{\mathbf{x},k|k-1,i} \alpha_{\mathbf{n},k,j} N\left(\mathbf{y}_k; \hat{\mathbf{y}}_{k|k-1,r}, \mathbf{P}_{yy,r}\right)}{\sum_{j=1}^{N_{\mathbf{n},k}} \sum_{i=1}^{N_{\mathbf{x},k}} \alpha_{\mathbf{x},k|k-1,i} \alpha_{\mathbf{n},k,j} N\left(\mathbf{y}_k; \hat{\mathbf{y}}_{k|k-1,r}, \mathbf{P}_{yy,r}\right)}$.

The mean $\hat{\mathbf{x}}_{k|k,r}$, covariance $\mathbf{P}_{k|k,r}$ as well as $\hat{\mathbf{y}}_{k|k-1,r}$ and $\mathbf{P}_{yy,r}$ can be obtained by Gaussian approximation filters, such as EKF or UKF.

CHAPTER III

SPARSE-GRID BASED NONLINEAR FILTERS

In this chapter, three sparse-grid based filters, including the sparse Gauss-Hermite quadrature (GHQ) filter, sparse-grid quadrature filter, and anisotropic sparse-grid quadrature filter, are introduced.

Before introducing this new filter, it is worth discussing the motivation of using the sparse-grid method.

Take one dimensional Gaussian type integral as an example. Two different types of integrals $\int_{\mathbb{R}} f(x)N(x;0,1)dx$ and $\int_{\mathbb{R}} f(x)^2 N(x;0,1)dx$ are needed to be calculated in the filtering algorithm.

By the Taylor series expansion, $f(x)$ can be represented by

$$f(x) \approx f(0) + f'(0)(x-0) + \frac{f''(0)}{2!}(x-0)^2 + \frac{f'''(0)}{3!}(x-0)^3 + \cdots \tag{3.1}$$

Hence, the following two equations can be obtained.

$$E[f(x)] \approx f(0) + f'(0)E[x] + \frac{f''(0)}{2!}E[x^2] + \frac{f'''(0)}{3!}E[x^3] + \cdots \tag{3.2}$$

$$E[f(x)f(x)] \approx (f(0))^2 + (f'(0))^2 E[x^2] + \left(\frac{f''(0)}{2!}\right)^2 E[x^4] + \left(\frac{f'''(0)}{3!}\right)^2 E[x^6] + \cdots \tag{3.3}$$

It can be seen from the above equations that more accurate expectation can be obtained by using more terms in the Taylor series expansion. In addition, the problem is converted

to how to calculate the high-order moments of x with a standard normal distribution. The conclusion is also true for the multidimensional Gaussian type integrals.

Recall that the UT and the cubature rule can capture up to third-order information of $f(x)$ when they are used to calculate the integrals. However, the higher-order information is discarded. To improve the accuracy, the GHQ rule is proposed and can capture higher-order information. Unfortunately, the GHQ rule has the *curse of dimensionality* problem. To alleviate the *curse of dimensionality* while maintaining close performance to GHQ, the sparse-grid quadrature rules and nonlinear filters are proposed as follows.

3.1 Sparse Gauss-Hermite Quadrature Filter

3.1.1 Sparse Gauss-Hermite Quadrature

In this section, the sparse Gauss-Hermite Quadrature (SGHQ) is introduced. The SGHQ is based on the Smolyak's rule for multivariate extension of the univariate GHQ rule and integration operators. The Smolyak rule is given by [41, 84, 85]

$$\int_{\mathbb{R}^n} \boldsymbol{f}(\mathbf{x}) N(\mathbf{x};\mathbf{0},\mathbf{I}_n)\, d\mathbf{x} \approx I_{n,L}(\boldsymbol{f}) \triangleq \sum_{\Xi \in \Upsilon_{n,L}} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_n}\right)(\boldsymbol{f}) \tag{3.4}$$

where $\mathbf{x} = \left[x_1, \cdots, x_j, \cdots, x_n\right]^T$; $\Xi \triangleq (i_1, \cdots, i_n)$ is the accuracy level sequence; $I_{n,L}(\boldsymbol{f})$ is the multidimensional integral of the function $\boldsymbol{f}$ with respect to a Gaussian distribution with the accuracy level L, $L \in \mathbb{N}$; $\mathbb{N}$ is the set of natural numbers, and n is the dimension. By accuracy level L, it means that $I_{n,L}(\boldsymbol{f})$ is exact for all polynomials of the form $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ with $\sum_{j=1}^{n} i_j \le 2L-1$ [56]. The operator "$\otimes$" denotes the tensor product and $\Delta^{i_j} = I_{i_j} - I_{i_j-1}$, $\forall i_j \in \mathbb{N}, (j = 1, \cdots, n)$, where $I_0 = 0$ and I_{i_j} is the univariate GHQ rule with the accuracy level-$i_j \in \Xi$.

$$I_{i_j} \triangleq \sum_{\gamma_{s_j} \in \mathbf{X}_{i_j}} f\left(\gamma_{s_j}\right) w_{s_j} \approx \int_{\mathbb{R}} f(x) N(x;0,1)\, dx \quad (s_j \in \mathbb{N}) \tag{3.5}$$

where X_{i_j} is the univariate GHQ point set with the accuracy level-i_j; γ_{s_j} and w_{s_j} are the univariate GHQ point and weight, respectively. By the accuracy level-i_j for the univariate GHQ rule, it means that I_{i_j} is exact to at least the $(2i_j - 1)^{\text{th}}$ order of all univariate polynomials. The minimum number of quadrature points for I_{i_j} is i_j [25].

The accuracy level set $\Upsilon_{n,L}$ in Eq. (3.4) is defined by [84]

$$\Upsilon_{n,L} \triangleq \left\{ \Xi \subset \mathbb{N}^n; i_j \geq 1, \sum_{j=1}^{n} \left(i_j - 1\right) \leq \left(L - 1\right) \right\} \tag{3.6}$$

where $\mathbb{N}^n$ denotes an n-element sequence of natural numbers.

In order to use the univariate quadrature rule, Eq. (3.4) can be rewritten as [84, 86]

$$I_{n,L}(\boldsymbol{f}) = \sum_{\Xi \in \mathbf{Y}_{n,L}} (-1)^{L-1+n-|\Xi|} \begin{pmatrix} n-1 \\ L-1+n-|\Xi| \end{pmatrix} \left(I_{i_1} \otimes \cdots I_{i_n}\right)(\boldsymbol{f}) \tag{3.7}$$

where $|\Xi|$ denotes the summation of the elements in Ξ;

$$\mathbf{Y}_{n,L} \triangleq \left\{ \Xi \subset \mathbb{N}^n : i_j \geq 1, L - n \leq \sum_{j=1}^{n} \left(i_j - 1\right) \leq \left(L - 1\right) \right\} \tag{3.8}$$

and

$$\left(I_{i_1} \otimes \cdots I_{i_n}\right)(\boldsymbol{f}) \triangleq \sum_{\gamma_{s_1} \in X_{i_1}} \cdots \sum_{\gamma_{s_n} \in X_{i_n}} w_{s_1} \cdots w_{s_n} \boldsymbol{f}\left(\gamma_{s_1}, \cdots, \gamma_{s_n}\right) \tag{3.9}$$

where X_{i_j} is the univariate point set of I_{i_j} and w_{s_j} is the corresponding weight of the point $\gamma_{s_j} \in X_{i_j}$.

Equation (3.7) is equivalent to the following equation that was used in Refs. [84, 86]

$$I_{n,L}(\boldsymbol{f}) = \sum_{q=L-n}^{L-1} (-1)^{L-1-q} \begin{pmatrix} n-1 \\ L-1-q \end{pmatrix} \sum_{\Xi \in \mathbf{N}_q^n} \left(I_{i_1} \otimes \cdots I_{i_n}\right)(\boldsymbol{f}) \tag{3.10}$$

where q is an auxiliary parameter to prescribe the range of the univariate accuracy level-i_j.

$\mathbf{N}_q^n$ in Eq. (3.10) is defined as [84]

$$\mathbf{N}_q^n \triangleq \begin{cases} \left\{ \Xi \subset \mathbb{N}^n : \sum_{j=1}^{n} i_j = n+q \right\} & \text{for } q \geq 0 \\ \varnothing & \text{for } q < 0 \end{cases} \tag{3.11}$$

The corresponding sparse-grid point set defined by $\mathbf{X}_{n,L}$ is given by

$$\mathbf{X}_{n,L} = \bigcup_{\Xi \in \mathbf{Y}_{n,L}} \left(X_{i_1} \otimes \cdots \otimes X_{i_n} \right) \tag{3.12}$$

The weight W_i for each multivariate GHQ point $\boldsymbol{\gamma}_i$ in $\mathbf{X}_{n,L}$ is the sum of the weights on the point over all combinations of $X_{i_1} \otimes \cdots \otimes X_{i_n}$ containing that point [25]. Moreover, for one specific combination Ξ, the weight on the point $\boldsymbol{\gamma}_i$ is calculated by $W_i = (-1)^{L-1+n-|\Xi|} \begin{pmatrix} n-1 \\ L-1+n-|\Xi| \end{pmatrix} \left(w_{s_1} \times \cdots \times w_{s_j} \times \cdots \times w_{s_n} \right)$ [25].

To better illustrate the SGHQ, we show how to use the Smolyak's rule to construct $X_{2,3}(n=2,\ L=3)$ from the univariate GHQ point sets X_1, X_2 and X_3 containing 1, 3, and 7 points, respectively, as shown in Figure 3.1. The single point in X_1 is represented by a circle „o"; the points in X_2 other than the origin are represented by asterisk „*"; and the points in X_3 other than the origin are represented by point „• ". Note that we use $2^L - 1$ points for the univariate point set with the accuracy level-L as an example. Other choices of point sets can be adopted as well and will be discussed in detail afterwards. From Eq. (3.10), q can be 1 or 2 and then we have $\mathbf{N}_1^2 = \{(1,2),(2,1)\}$ and $\mathbf{N}_2^2 = \{(1,3),(2,2),(3,1)\}$. The first combination in $\mathbf{N}_1^2$ is $(1,2)$, i.e. $i_1 = 1$ and $i_2 = 2$. This determines the tensor product $X_1 \otimes X_2$ and leads to the 3 points along the horizontal axis. Similarly, the second combination in $\mathbf{N}_1^2$ is $(2,1)$, i.e. $i_1 = 2$ and $i_2 = 1$, which generates the tensor product $X_2 \otimes X_1$ and 3 points along the vertical axis. Note that in the

tensor products of $X_1 \otimes X_2$ and $X_2 \otimes X_1$, there are two new points added to the point sets along the two axes and one repeated point at the origin contributed from X_1, which is considered as an old point. Other tensor products in $\mathbf{N}_2^2$ can be generated in the same way. The final sparse-grid point set $X_{2,3}$ leads to 21 points as shown on the bottom right of Figure 3.1. In comparison, the point set of the conventional GHQ rule is only determined by $X_3 \otimes X_3$ generated from the 7-point X_3, which results in 49 points as shown on the upper right of Figure 3.1. As can be seen, the SGHQ uses significantly fewer points than the conventional GHQ.

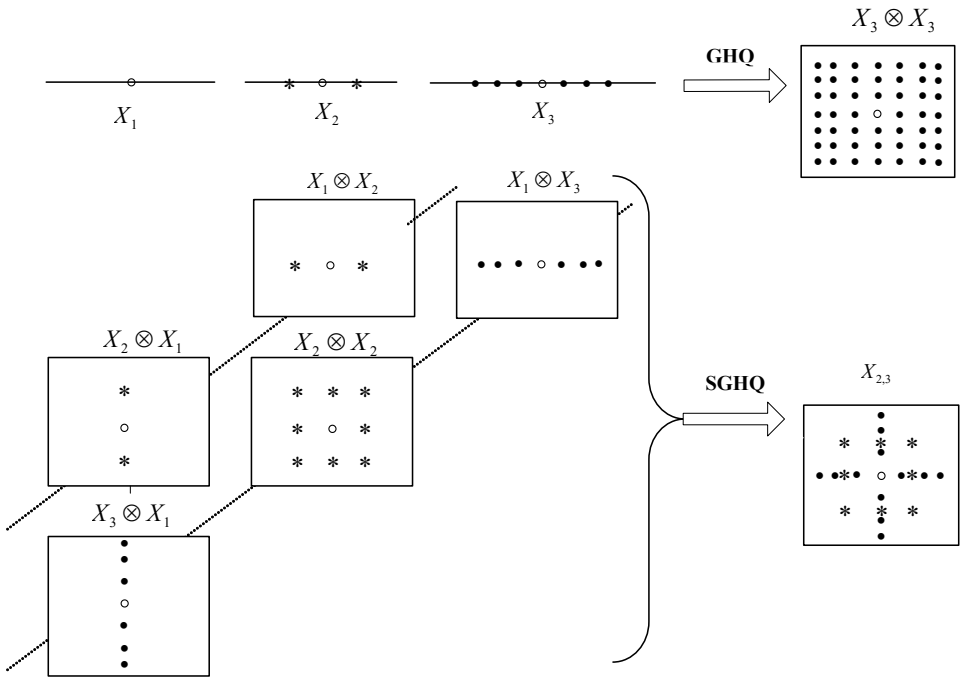

Figure 3.1 Multivariate GHQ and SGHQ with the accuracy level-3 for 2-dimensional problems

The following theorem reveals the relationship between the univariate GHQ and the multivariate SGHQ in terms of the accuracy.

Theorem 3.1([85]): Assume that the sequence of univariate quadrature rules $I = \{I_i : i \in \mathrm{N}\}$ is defined such that I_i is exact for all univariate polynomials of order up to $2i-1$. Then, the Smolyak rule $I_{n,L}$ using I as the univariate basis sequence is exact for *n*-variate polynomials of total order up to $2L-1$.

Proof: Refer to [85].

Remark 3.1: In Figure 3.1, 2^L-1 points are used for the level-*L* univariate GHQ. The number of points other than 2^L-1 can be used as well, for example, L (minimum number of points) or $2L-1$. Different univariate point-selection strategies generate different locations of the univariate GHQ points, which is similar to the effect of the tunable parameter κ for the UKF. Increasing the number of univariate GHQ points can increase the accuracy level for that particular dimension. It is important to note that increasing the accuracy level of the univariate GHQ may or may not increase the accuracy level of the multivariate SGHQ.

The generation of SGHQ points and weights is given by Algorithm I. Note that some of the weights may be negative, which is the case for the UT as well.

Algorithm 3.1 Generate SGHQ Points and Weights

$[\chi, W] = \text{SGHQ}[n, L]$

(χ : SGHQ point set; W: weight set with the element of w_s)

FOR $q = L-n : L-1$

Determine $\mathbf{N}_q^n$

FOR each element $\Xi=(i_1,\cdots,i_n)$ in $\mathbf{N}_q^n$, form $X_{i_1} \otimes X_{i_2} \cdots \otimes X_{i_n}$

FOR each point in $X_{i_1} \otimes X_{i_2} \cdots \otimes X_{i_n}$

IF the point is new, add it to χ, assign a new index s to the point and calculate its weight as

$$w_s = (-1)^{L-1-q} \mathbf{C}_{n-1}^{L-1-q} \prod_{p=1}^{n} w_{i_p} \tag{3.13}$$

where $\mathbf{C}_{n-1}^{L-1-q}$ is the binomial operator.

ELSE (the point is already existing) update the old weight by

$$w_s = w_s + (-1)^{L-1-q} \mathbf{C}_{n-1}^{L-1-q} \prod_{p=1}^{n} w_{i_p} \tag{3.14}$$

(w_{i_p} is the GHQ weight for the univariate x_p, one of the state variables; w_{i_p} can be calculated by the univariate GHQ rule in Section 2.2.1.5)

END IF

END FOR

END FOR

END FOR

The following theorem reveals the relationship between the SGHQ and the UT, and shows that the UKF with certain parameters is a subset of the SGHQF.

Theorem 3.2: If one point and three points are used for the univariate level-1 and level-2 GHQ respectively, the UT with the suggested optimal parameter $\kappa = 3 - n$ is identical to the SGHQ with the accuracy level-2; if one point and two points are used for the univariate level-1 and level-2 GHQ respectively, the UT with $\kappa = 1 - n$ is identical to the SGHQ with the accuracy level-2.

Proof: If one point and three points are used for the univariate level-1 and level-2 GHQ respectively, using the univariate GHQ rule discussed in Section 2.2.1.5, the point set at level-1 is chosen as $X_1 = \{0\}$ with the weight of 1. At level-2, the univariate 3-point set is $X_2 = \left\{0, \sqrt{3}, -\sqrt{3}\right\}$ with the corresponding weight sequence of $\left(\frac{2}{3}, \frac{1}{6}, \frac{1}{6}\right)$. For the accuracy level-2 (L=2), the value of q in $\mathbf{N}_q^n$ can be 0 or 1. $\mathbf{N}_0^n$ contains only one set Ξ, of which all elements are 1. The only point corresponding to $\mathbf{N}_0^n$ is $[0,0,\cdots 0,0]^T$ weighted by $(-1)^{2-1-0} \times \mathbf{C}_{n-1}^{2-1-0} \times \overbrace{1 \times 1 \times \cdots \times 1}^{n \text{ elements}} = -(n-1)$ (Eq. (3.13)). $\mathbf{N}_1^n$ consists of n sequences, i.e.

$$\mathbf{N}_1^n = \underbrace{\left\{ \overbrace{(2,1,\cdots 1,1)}^{n}, \overbrace{(1,2,\cdots 1,1)}^{n}, \cdots \overbrace{(1,\cdots 1,2,1,\cdots 1)}^{n}, \cdots \overbrace{(1,1,\cdots 1,2)}^{n} \right\}}_{n \text{ element set}}.$$

In each sequence, only one element equals 2 and all the others equal 1. Each element set in $\mathbf{N}_1^n$ determines a tensor product sequence of X_{i_j} where $i_j \in \Xi \in \mathbf{N}_1^n$. Take the combination $(2,1,\cdots 1,1)$ as an example. The sequence $(2,1,\cdots 1,1)$ implies the tensor product $X_2 \otimes X_1 \cdots \otimes X_1 \otimes X_1$. Because X_2 contains three one-dimensional points, the point set of $X_2 \otimes X_1 \cdots \otimes X_1 \otimes X_1$ contains three n-dimensional points $[0,0,\cdots 0,0]^T$, $\left[\sqrt{3},0,\cdots 0,0\right]^T$ and $\left[-\sqrt{3},0,\cdots 0,0\right]^T$ as well, with the respective weights (Eq. (3.10)),

$$(-1)^{2-1-1}\times \mathbf{C}_{n-1}^{2-1-1}\times \underbrace{\left(\frac{2}{3}\times \overbrace{1\times 1\times\cdots\times 1}^{n-1 \text{ elements}}\right)}_{\text{product of weights}}=\frac{2}{3},\quad (-1)^{2-1-1}\times \mathbf{C}_{n-1}^{2-1-1}\times \underbrace{\left(\frac{1}{6}\times \overbrace{1\times 1\times\cdots\times 1}^{n-1 \text{ elements}}\right)}_{\text{product of weights}}=\frac{1}{6}, \text{ and}$$

$$(-1)^{2-1-1}\times \mathbf{C}_{n-1}^{2-1-1}\times \underbrace{\left(\frac{1}{6}\times \overbrace{1\times 1\times\cdots\times 1}^{n-1 \text{ elements}}\right)}_{\text{product of weights}}=\frac{1}{6}.$$

Since the point $[0,0,\cdots 0,0]^T$ appears n+1 times (n times from $\mathbf{N}_1^n$ and one time from $\mathbf{N}_0^n$) while the other points appear only once, the final weight of $[0,0,\cdots 0,0]^T$ is calculated as a sum (Eq. (3.14)):

$$(-1)^{2-1-0}\times \mathbf{C}_{n-1}^{2-1-0}\times \underbrace{\left(\overbrace{1\times 1\times\cdots\times 1}^{n-1 \text{ elements}}\times 1\right)}_{\text{product of weights}}+(-1)^{2-1-1}\times \mathbf{C}_{n-1}^{2-1-1}\times$$

$$\left\{\overbrace{\underbrace{\left(\frac{2}{3}\times \overbrace{1\times 1\times\cdots\times 1}^{n-1 \text{ elements}}\right)}_{\text{product of weights}}+\underbrace{\left(1\times \frac{2}{3}\times \overbrace{1\times 1\times\cdots\times 1}^{n-2 \text{ elements}}\right)}_{\text{product of weights}}+\cdots+\underbrace{\left(\overbrace{1\times 1\times\cdots\times 1}^{n-1 \text{ elements}}\times \frac{2}{3}\right)}_{\text{product of weights}}}^{n \text{ elements}}\right\}$$

Simplifying it, we get $-(n-1)+\frac{2n}{3}=\frac{3-n}{3}$.

To summarize, the points $\boldsymbol{\gamma}_i$ and the weights W_i of the SGHQ with the accuracy level-2 are

$$\begin{cases} \boldsymbol{\gamma}_1=[0,0,\cdots 0,0]^T \\ \boldsymbol{\gamma}_i=\sqrt{3}\mathbf{e}_{i-1}, \quad 2\le i\le n+1 \\ \boldsymbol{\gamma}_i=-\sqrt{3}\mathbf{e}_{i-n-1}, \quad n+2\le i\le 2n+1 \end{cases} \quad \begin{cases} W_1=\frac{3-n}{3}, \\ W_i=\frac{1}{6} \quad (i=2\cdots 2n+1) \end{cases} \tag{3.15}$$

They are exactly the same as the UT if $\kappa=3-n$ (comparing Eq. (3.15) with Eq. (2.46)).

If one point and two points are used for the univariate level-1 and level-2 GHQ respectively, the corresponding point sets become $X_1=\{0\}$ with the weight of 1 and $X_2=\{1,-1\}$ with the weight sequence of $\left(\frac{1}{2},\frac{1}{2}\right)$. For the accuracy level-2, the value of q in $\mathbf{N}_q^n$ can be 0 or 1. $\mathbf{N}_0^n$ contains only one sequence, of which all elements are 1. The

only point corresponding to $\mathbf{N}_0^n$ is $[0,0,\cdots 0,0]^T$ weighted by $(-1)^{2-1-0}\times \mathbf{C}_{n-1}^{2-1-0}\times \overbrace{1\times 1\times\cdots\times 1}^{n \text{ elements}} = -(n-1) = 1-n$. $\mathbf{N}_1^n$ consists of n sequences. In each sequence, only one element equals 2 and all the others equal 1. Consider the combination $(2,1,\cdots 1,1)$ as an example. Because X_2 contains two one-dimensional points, the point set $X_2 \otimes X_1 \cdots \otimes X_1$ corresponding to $(2,1,\cdots 1,1)$ contains two n-dimensional points $[1,0,\cdots 0,0]^T$ and $[-1,0,\cdots 0,0]^T$ as well, with the respective weights (Eq. (3.13)) $(-1)^{2-1-1}\times \mathbf{C}_{n-1}^{2-1-1}\times \frac{1}{2}\times \overbrace{1\times 1\times\cdots\times 1}^{n-1 \text{ elements}} = \frac{1}{2}$ and $(-1)^{2-1-1}\times \mathbf{C}_{n-1}^{2-1-1}\times \frac{1}{2}\times \overbrace{1\times 1\times\cdots\times 1}^{n-1 \text{ elements}} = \frac{1}{2}$. To summarize, the points $\boldsymbol{\gamma}_i$ and the weights W_i of the SGHQ for this case are

$$\begin{cases} \boldsymbol{\gamma}_1 = [0,0,\cdots 0,0]^T \\ \boldsymbol{\gamma}_i = \mathbf{e}_{i-1}, \quad 2 \le i \le n+1 \\ \boldsymbol{\gamma}_i = -\mathbf{e}_{i-n-1}, \quad n+2 \le i \le 2n+1 \end{cases} \quad \begin{cases} W_1 = 1-n \\ W_i = \frac{1}{2} \quad (i = 2\cdots 2n+1) \end{cases} \tag{3.16}$$

They are exactly the same as the UT if $\kappa = 1-n$ (comparing Eq. (3.16) with Eq. (2.46)). Note that in this case, there is no repeated/old point. Therefore, Eq. (3.14) is not used. ■

Proposition 3.1 Given the accuracy level-L and $L-1 \le n$, the number of points for the SGHQ increases polynomially with the dimension n and the highest order of this polynomial is $L-1$.

Proof: When $L \le n$, q can be $0,1,\cdots,L-1$. Thus, $\mathbf{N}_q^n$ can be $\mathbf{N}_0^n,\cdots,\mathbf{N}_{L-1}^n$. When $L = n+1$, q can be $1,\cdots,L-1$. For $\mathbf{N}_q^n$, each element sequence in it contains at most q elements that are greater than 1. If it contains $q+1$ elements greater than 1, then we have $\sum_{d=1}^{n} i_d \ge \underbrace{n-q-1}_{\text{element=1}} + \underbrace{2\times(q+1)}_{\text{element =2}} > n+q$, which contradicts the definition of $\mathbf{N}_q^n$. The largest q for $\mathbf{N}_q^n$ to be a nonempty set is L-1.

When $q = L-1$, each element sequence in $\mathbf{N}_{L-1}^n$ contains at most $L-1$ elements greater than 1. If there are $L-1$ elements greater than 1, there are $\mathbf{C}_n^{L-1}$ combinations of

accuracy level sequences. If there are p $(1 \le p < L-1)$ elements greater than 1, there are $\mathbf{C}_n^p$ combinations of such accuracy level sequences.

When $0 \le q < L-1$, each element sequence in $\mathbf{N}_q^n$ contains at most q elements greater than 1. If there are q elements greater than 1, there are $\mathbf{C}_n^q$ combinations of such accuracy level sequences. If there are p $(1 \le p < q)$ elements greater than 1, there are $\mathbf{C}_n^p$ combinations of such accuracy level sequences.

For each accuracy level sequence, the number of new points is bounded and is not equal to 0 since there is at most one point shared by different univariate GHQ rules. The total number of points is a linear combination of the product of the number of new points generated from each accuracy level sequence and the associated combinatorial $\mathbf{C}_n^t$ (t=p or q), which is a polynomial in terms of the dimension n. Hence, when the accuracy level L is given, the total number of points is a polynomial in terms of n and the highest order of this polynomial is $L-1$, which is determined by $\mathbf{C}_n^{L-1}$.■

Remark 3.2: In general, the accuracy level L is significantly smaller than the dimension n. High accuracy level SGHQ needs more points and increases the computation load unnecessarily when a low accuracy level can provide acceptable performance.

For convenience, we give the general formula for calculating the number of SGHQ points with the accuracy level-2 and level-3. Assume the univariate GHQ with the level-L contains m_L points. $\mathbf{N}_0^n = \{(1,\cdots,1)\}$ generates one point (origin point). $\mathbf{N}_1^n = \{(2,1,\cdots,1)\cdots(1,1,\cdots,2)\}$ generates $m_2^* \times \mathbf{C}_n^1$ new points (excluding the origin point), where $m_2^* = \begin{cases} m_2 - 1; & m_2 \text{ is odd} \\ m_2; & m_2 \text{ is even} \end{cases}$. That is because univariate GHQs with an odd number of points share the midpoint.

For $\mathbf{N}_2^n = \left\{\left\{(3,1,\cdots,1),\cdots,(1,1,\cdots,3)\right\},\left\{(2,2,1,\cdots,1),\cdots,(1,\cdots,2,\cdots,2,\cdots 1)\cdots,(1,\cdots 1,2,2)\right\}\right\}$, the first part of $\mathbf{N}_2^n$ generates $m_3^* \times \mathbf{C}_n^1$ new points (excluding the origin point), where $m_3^* = \begin{cases} m_3 - 1; & m_3 \text{ is odd} \\ m_3; & m_3 \text{ is even} \end{cases}$. The second part of $\mathbf{N}_2^n$ generates $m^* \times \mathbf{C}_n^2$ new points (excluding the origin point). $m^* = \begin{cases} (m_2 - 1)^2; & m_2 \text{ is odd} \\ (m_2)^2; & m_2 \text{ is even} \end{cases}$.

For the level-2, when $n = 1$, the SGHQ is not necessary. When $n \geq 2$, $q = 0,1$. When $q = 0$, $\mathbf{N}_0^n = \left\{(1,\cdots,1)\right\}$ generates one point (origin point). When $q = 1$, $\mathbf{N}_1^n = \left\{(2,1,\cdots,1)\cdots(1,1,\cdots,2)\right\}$ generates $m_2^* \times \mathbf{C}_n^1$ points (excluding the origin point). Hence, for the level-2, the total number of points is $m_2^* \times \mathbf{C}_n^1 + 1 = m_2^* n + 1$.

For the level-3, when $n = 1$, the SGHQ is not necessary. When $n = 2$, $q = 1$ or 2. When $q = 1$, $\mathbf{N}_1^n$ generates $m_2^* \times \mathbf{C}_n^1$ new points (excluding the origin point). When $q = 2$, $\mathbf{N}_2^n$ generates $m^* \times \mathbf{C}_n^2 + m_3^* \times \mathbf{C}_n^1$ new points (excluding the origin point). Whether the origin point exists in the final point set depends on the value of m_2 and m_3, which can be summarized as

$$\begin{cases} m^* \times \mathbf{C}_n^2 + \left(m_3^* + m_2^*\right) \times \mathbf{C}_n^1 + 1 = m^* \dfrac{n(n-1)}{2} + \left(m_3^* + m_2^*\right) n + 1 & \text{if } m_2 \text{ or } m_3 \text{ is odd number} \\ m^* \times \mathbf{C}_n^2 + \left(m_3^* + m_2^*\right) \times \mathbf{C}_n^1 = m^* \dfrac{n(n-1)}{2} + \left(m_3^* + m_2^*\right) n & \text{if } m_2 \text{ and } m_3 \text{ are even number} \end{cases}.$$

When $n \geq 3$, $q = 0,\ 1,\ 2$. When $q = 0$, $\mathbf{N}_0^n$ generates the origin point. When $q = 1$, $\mathbf{N}_1^n$ generates $m_2^* \times \mathbf{C}_n^1$ new points. When $q = 2$, $\mathbf{N}_2^n$ generates the same new points as $n = 2$, i.e. $m^* \times \mathbf{C}_n^2 + m_3^* \times \mathbf{C}_n^1$.

Hence, for the level-3($n \geq 3$), the total number of points is

$$m^* \times \mathbf{C}_n^2 + \left(m_3^* + m_2^*\right) \times \mathbf{C}_n^1 + 1 = m^* \frac{n(n-1)}{2} + \left(m_3^* + m_2^*\right) n + 1$$

Based on the above discussions, for different univariate GHQ point-selection strategies, the number of multivariate SGHQ points with accuracy level-2 and level-3 is summarized in Table 3.1.

Remark 3.3: The points for the univariate GHQ rule are symmetric. Thus, the points corresponding to each accuracy level sequence in $\mathbf{N}_q^n$ are symmetric as well because they are generated by the tensor product of different level univariate GHQ rules. Hence, the points generated by the SGHQ rule are symmetric.

Table 3.1 Number of SGHQ points for different univariate GHQ point-selection strategies with the accuracy level-2 and level-3

m_L \ SGHQ	level-2 (L=2, $n \geq 2$)	level-3 (L=3, $n \geq 3$)
L	$2n+1$	$2n^2+2n+1$
$2L-1$	$2n+1$	$2n^2+4n+1$
2^L-1	$2n+1$	$2n^2+6n+1$

For better illustration, the point sets with the accuracy level-2 (using 3-point univariate GHQ) and level-3 (using 7-point univariate GHQ) for 2-dimensional and 3-dimensional problems are shown in Figure 3.2 and Figure 3.3 respectively.

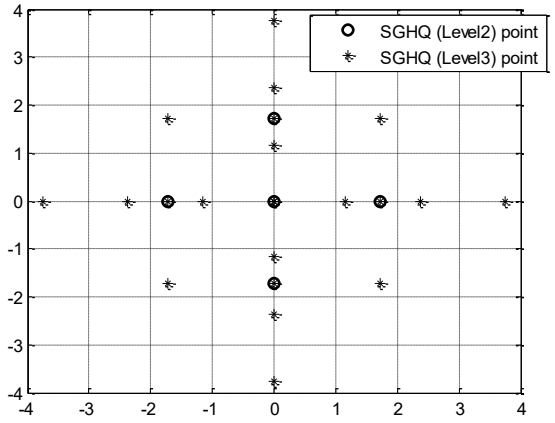

Figure 3.2 Point set illustration for 2-dimensional SGHQ (level-2 and level-3)

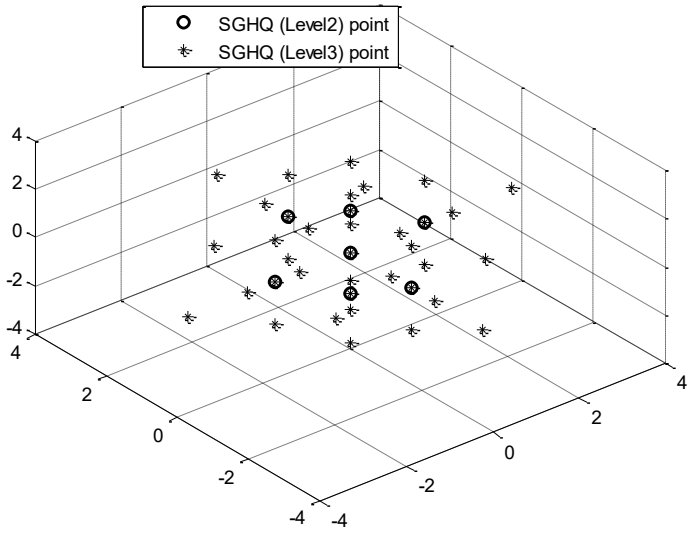

Figure 3.3 Point set illustration for 3-dimensional SGHQ (level-2 and level-3)

It has been shown that SGHQ (level-2) when using l points for univariate GHQ rule I_l is identical with the UT with $\kappa = 1 - n$ and SGHQ (level-2) when using $2l$-1 points for univariate GHQ rule I_l is identical with the UT with $\kappa = 3 - n$. For convenience, the formula of the SGHQ (level-3), $I_{n,3}(\boldsymbol{f})$, using different point-selection strategies, is derived as follows. Two different cases are considered.

In the first case, l points are used for univariate GHQ rule I_l. The points and weights for different I_l are listed in Table 3.2.

Table 3.2 Univariate GHQ points and weights of the first case

	Points	weights
I_1	0	1
I_2	-1, 1	1/2, 1/2
I_3	$-\sqrt{3}$, 0, $\sqrt{3}$	1/6, 2/3, 1/6

By Eq. (3.10), q can be 0, 1, or 2. When $q=0$, $\mathbf{N}_0^n=[1,\cdots,1]^T$. This leads to one point $[0,\cdots,0]^T$ with weight $(-1)^2\mathbf{C}_{n-1}^2=(n-1)(n-2)/2$.

When $q=1$, one element in $\mathbf{N}_1^n$ is 2 and others are 1. This leads to $2n$ points $P_1^{(i)}$ with weight A_1

$$P_1^{(i)}=\begin{cases}\mathbf{e}_i & i=1,\cdots,n\\ -\mathbf{e}_{i-n} & i=n+1,\cdots,2n\end{cases} \tag{3.17}$$

$$A_1=(-1)^1\mathbf{C}_{n-1}^1\left(\frac{1}{2}\times\overbrace{1\cdots\times1}^{n-1\text{ elements}}\right)=-\frac{n-1}{2} \tag{3.18}$$

When $q=2$, elements of Ξ in $\mathbf{N}_2^n$ have two kinds of combinations. For the first kind, one element in $\mathbf{N}_2^n$ is 3 and others are 1. For the second kind, two elements in $\mathbf{N}_2^n$ are 2 and others are 1. The first kind leads to points $P_2^{(i)}$ with weight A_2 and $[0,\cdots,0]^T$ with weight $2/3\times1\times n=2n/3$, i.e.

$$P_2^{(i)}=\begin{cases}\sqrt{3}\mathbf{e}_i & i=1,\cdots,n\\ -\sqrt{3}\mathbf{e}_{i-n} & i=n+1,\cdots,2n\end{cases} \tag{3.19}$$

$$A_2=(-1)^0\mathbf{C}_{n-1}^0\left(\frac{1}{6}\times\overbrace{1\cdots\times1}^{n-1\text{ elements}}\right)=\frac{1}{6} \tag{3.20}$$

Similarly, the second kind leads to $P_3^{(i)}$ with weight A_3.

$$P_3^{(i)} = \begin{cases} \mathbf{e}_j + \mathbf{e}_k & j,k = 1,\cdots n; j < k \\ \mathbf{e}_j - \mathbf{e}_k & j,k = 1,\cdots n; j < k \\ -\mathbf{e}_j + \mathbf{e}_k & j,k = 1,\cdots n; j < k \\ -\mathbf{e}_j - \mathbf{e}_k & j,k = 1,\cdots n; j < k \end{cases} \tag{3.21}$$

$$A_3 = (-1)^0 \mathbf{C}_{n-1}^0 \left(\frac{1}{2} \times \frac{1}{2} \times \overbrace{1 \cdots \times 1}^{n-2 \text{ elements}} \right) = \frac{1}{4} \tag{3.22}$$

By Eq. (3.14), the weight A_0 corresponding to the point $[0,\cdots,0]^T$ can be calculated by

$$A_0 = -\frac{(n-1)(n-2)}{2} + \frac{2n}{3} = -\frac{3n^2 - 5n + 6}{6} \tag{3.23}$$

Based on Eq. (3.10), the following equation can be obtained.

$$I_{n,3}(\boldsymbol{f}) = A_0 \boldsymbol{f}(\mathbf{0}) + A_1 \sum_{i=1}^{2n} \boldsymbol{f}\left(P_1^{(i)}\right) + A_2 \sum_{i=1}^{2n} \boldsymbol{f}\left(P_2^{(i)}\right) + A_3 \sum_{i=1}^{4\mathbf{C}_n^2} \boldsymbol{f}\left(P_3^{(i)}\right) \tag{3.24}$$

Note, the number of points used in Eq. (3.24) is $2n^2 + 2n + 1$. In addition, it can be verified that the summation of all weights in Eq. (3.24) is 1.

In the second case, $2l-1$ points are used for the univariate GHQ rule I_l. The points and weights for different accuracy univariate GHQ rules are listed in Table 3.3.

Table 3.3 Univariate GHQ points and weights of the second case

	points	weights	weight values
I_1	0	w_0	1
I_2	0	w_1	2/3
	$\pm\sqrt{3}$	w_2	1/6
I_3	0	w_3	0.53333
	$\pm\sqrt{5-\sqrt{10}}$	w_4	0.22208
	$\pm\sqrt{5+\sqrt{10}}$	w_5	0.01126

For the accuracy level L=3, by Eq. (3.10), q can be 0, 1, or 2. When $q=0$, by Eq. (3.11), $\mathbf{N}_0^n=[1,\cdots,1]^T$. This leads to one point $[0,\cdots,0]^T$ with the weight $(-1)^2\mathbf{C}_{n-1}^2=(n-1)(n-2)/2$ calculated by Eq. (3.13).

When $q=1$, one element of Ξ in $\mathbf{N}_1^n$ is 2 and others are 1. This leads to points $P_1^{(i)}$ with weight $(-1)^1\mathbf{C}_{n-1}^1(w_2\times 1)$ calculated by Eq. (3.13). and n duplicated points $[0,\cdots,0]^T$ with weight $(-1)^1\mathbf{C}_{n-1}^1(w_1\times 1)$ calculated by Eq. (3.13).

$$P_1^{(i)}=\begin{cases}\sqrt{3}\mathbf{e}_i & i=1,\cdots 2n\\ -\sqrt{3}\mathbf{e}_{i-n}\end{cases} \tag{3.25}$$

When $q=2$, the elements of Ξ in $\mathbf{N}_2^n$ have two possible combinations. For the first combination, one element of Ξ is 3 and others are 1. For the second combination, two elements of Ξ are 2 and others are 1. The first combination leads to points $P_2^{(i)}$ with weight A_2, points $P_3^{(i)}$ with weight A_3, and n duplicated points $[0,\cdots,0]^T$ with weight $w_3\times 1=w_3$ where

$$P_2^{(i)}=\begin{cases}\sqrt{5-\sqrt{10}}\mathbf{e}_i & i=1,\cdots 2n\\ -\sqrt{5-\sqrt{10}}\mathbf{e}_{i-n}\end{cases} \tag{3.26}$$

$$A_2=w_4 \tag{3.27}$$

$$P_3^{(i)}=\begin{cases}\sqrt{5+\sqrt{10}}\mathbf{e}_i & i=1,\cdots 2n\\ -\sqrt{5+\sqrt{10}}\mathbf{e}_{i-n}\end{cases} \tag{3.28}$$

$$A_3=w_5 \tag{3.29}$$

Similarly, the second combination leads to points $P_4^{(i)}$ with weight A_4, $n-1$ duplicated $P_1^{(i)}$ with weight $w_1\times w_2$, and $n(n-1)/2$ duplicated $[0,\cdots,0]^T$ with weight w_1^2.

$$P_4^{(i)}=\begin{cases}\sqrt{3}(\mathbf{e}_j+\mathbf{e}_k) & j,k=1,\cdots n;\ j<k\\ \sqrt{3}(\mathbf{e}_j-\mathbf{e}_k) & j,k=1,\cdots n;\ j<k\\ \sqrt{3}(-\mathbf{e}_j+\mathbf{e}_k) & j,k=1,\cdots n;\ j<k\\ \sqrt{3}(-\mathbf{e}_j-\mathbf{e}_k) & j,k=1,\cdots n;\ j<k\end{cases} \tag{3.30}$$

$$A_4 = w_2^2 \tag{3.31}$$

Since the point $[0,\cdots,0]^T$ is duplicated in all cases, by Eq. (3.14), the final weight of point $[0,\cdots,0]^T$, denoted by A_0, can be calculated by

$$A_0 = 0.5(n-1)\left(n-2+nw_1^2\right)-n(n-1)w_1+nw_3 \tag{3.32}$$

Similarly, the final weight of point $P_1^{(i)}$, denoted by A_1, can be calculated by

$$A_1 = -(n-1)w_2+(n-1)w_1w_2 \tag{3.33}$$

Hence, $I_{n,3}(\boldsymbol{f})$ can be obtained.

$$\begin{aligned} I_{n,3}(\boldsymbol{f}) = A_0\boldsymbol{f}(\mathbf{0}) + A_1\sum_{i=1}^{2n}\boldsymbol{f}\left(P_1^{(i)}\right) + A_2\sum_{i=1}^{2n}\boldsymbol{f}\left(P_2^{(i)}\right) \\ + A_3\sum_{i=1}^{2n}\boldsymbol{f}\left(P_3^{(i)}\right) + A_4\sum_{i=1}^{n(n-1)/2}\boldsymbol{f}\left(P_4^{(i)}\right) \end{aligned} \tag{3.34}$$

The sparse Gauss-Hermite quadrature filter can be obtained by using the SGHQ points and weights in the framework of the point-based Gaussian approximation filters (Section 2.2.1.2).

3.2 Sparse-grid Quadrature Filter

In this section, a more general sparse-grid quadrature (SGQ) filter is proposed based on the moment matching method that can include GHQ and UT as a special case and is more flexible than SGHQ.

To generate an *n*-dimensional sparse-grid quadrature (SGQ) with accuracy level-*L*, the following two steps are needed:

Step 1: *choose the univariate quadrature point sets and weights with accuracy levels from 1 to L using the moment matching method*.

Step 2: *extend the univariate quadrature point sets and weights to the n-dimensional SGQ points and weights using the sparse-grid method, i.e. Eq. (3.4)*.

3.2.1 Univariate Quadrature Points and Weights

For the *n*-dimensional sparse-grid quadrature (SGQ) with the accuracy level-*L* to be exact for all multivariate polynomials with the total degree of 2*L*-1, the univariate quadrature rule I_{i_j} ($i_j \in \Xi$ and $1 \le i_j \le L$) needs to be exact for all univariate polynomials of the degree up to $2i_j$-1, which can be satisfied using the following moment matching method.

The general moment matching formula used here (one-dimensional Gaussian type integral) is

$$M_j = \int_{-\infty}^{\infty} x^j N(x;0,1)\,dx = \sum_{i=1}^{N_u} \bar{w}_i \left(\bar{p}_i\right)^j \tag{3.35}$$

where M_j is the j^{th} order moment; N_u is the number of univariate quadrature points; $\bar{p}_i$ and $\bar{w}_i$ are the univariate quadrature points and weights, respectively. We require that the quadrature rule be exact for all univariate polynomials of order up to $N_u - 1$. Hence, $\bar{p}_i$ and $\bar{w}_i$ should satisfy the following equation:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \bar{p}_1 & \bar{p}_2 & \cdots & \bar{p}_{N_u} \\ \vdots & \vdots & & \vdots \\ \bar{p}_1^{N_u-1} & \bar{p}_2^{N_u-1} & \cdots & \bar{p}_{N_u}^{N_u-1} \end{bmatrix} \begin{bmatrix} \bar{w}_1 \\ \bar{w}_2 \\ \vdots \\ \bar{w}_{N_u} \end{bmatrix} = \begin{bmatrix} M_0 \\ M_1 \\ \vdots \\ M_{N_u-1} \end{bmatrix} \tag{3.36}$$

Note that a normalization constraint exists since $\sum_{i=1}^{N_u} \bar{w}_i = M_0 = 1$.

The values of $\bar{p}_i$ and $\bar{w}_i$ $(i = 1 \cdots N_u)$ can be obtained by solving the nonlinear equations (3.36). But the solution is not unique in general. Note that when we choose distinct points $\bar{p}_i$ $(i = 1 \cdots N_u)$, the coefficient matrix is the Vandermonde"s matrix whose determinant is nonzero and the weights $\bar{w}_i$ $(i = 1 \cdots N_u)$ can be uniquely determined. The univariate GHQ rule is obtained when $\bar{p}_i$ are chosen to be the roots of Hermite polynomials [29]. Because the N_u-point univariate GHQ is exact for polynomials of

order up to $(2N_u - 1)$ as mentioned in Section 2.2.1.5, the GHQ points can match up to the $(2N_u - 1)^{th}$ moments.

In this section, we use $N_u = 2l - 1$ symmetric points about the origin for the univariate quadrature point set with the accuracy level-*l*. The locations of the points are treated as free parameters and the weights are uniquely determined from the point locations. Other point selection methods are also possible, which will be discussed afterwards. Because symmetric points are used, we use different notations from those in Eq. (3.36) to represent the points ($\tilde{p}_i$) and weights ($\tilde{w}_i$). We also assume that the weights for $\tilde{p}_i$ and $-\tilde{p}_i$ are both $\tilde{w}_i$. Using this point-selection strategy, Eqs. (3.19) and (3.20) can be rewritten as Eqs. (3.37) and (3.38), respectively.

$$M_j = \begin{cases} \tilde{w}_1 + \sum_{i=2}^{((N_u-1)/2)+1} 2\tilde{w}_i & j=0 \\ 0 & j \text{ is odd} \\ \tilde{w}_1 0^j + \sum_{i=2}^{((N_u-1)/2)+1} 2\tilde{w}_i \tilde{p}_i^j & j \text{ is even} \end{cases} \tag{3.37}$$

Because $N_u = 2l - 1$, we have

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0^2 & \tilde{p}_2^2 & \cdots & \tilde{p}_l^2 \\ \vdots & \vdots & & \vdots \\ 0^{2l-2} & \tilde{p}_2^{2l-2} & \cdots & \tilde{p}_l^{2l-2} \end{bmatrix} \begin{bmatrix} \tilde{w}_1 \\ 2\tilde{w}_2 \\ \vdots \\ 2\tilde{w}_l \end{bmatrix} = \begin{bmatrix} M_0 \\ M_2 \\ \vdots \\ M_{2l-2} \end{bmatrix} \tag{3.38}$$

In Eq. (3.38), only the even moments are considered because the symmetry of the points implies that the odd moments are matched automatically. If we choose 2*l*-1 symmetric and distinct points, the coefficient matrix of Eq (3.38) is a Vandermonde"s matrix. The weights $\tilde{w}_i$ $(i = 1, \cdots, l)$ can be uniquely determined since the inverse of the Vandermonde"s matrix has an analytical form [87].

In the following, we use the SGQ with L=3 as an example to illustrate the moment matching method. According to the Smolyak"s rule Eq. (3.4), or Eq. (3.10), we need the univariate quadrature rules at level-1 (I_1), level-2 (I_2), and level-3 (I_3). For the univariate quadrature rule I_1, the point set is chosen to be $\{0\}$ with the corresponding weight of 1. For the univariate quadrature rule I_2, we choose the symmetric univariate quadrature point set as $\{-\hat{p}_1, 0, \hat{p}_1\}$ with the corresponding weight sequence $(\hat{w}_2, \hat{w}_1, \hat{w}_2)$. For the univariate quadrature rule I_3, we choose the symmetric univariate quadrature point set as $\{-\hat{p}_3, -\hat{p}_2, 0, \hat{p}_2, \hat{p}_3\}$ with the corresponding weight sequence $(\hat{w}_5, \hat{w}_4, \hat{w}_3, \hat{w}_4, \hat{w}_5)$. To satisfy the condition of *Theorem 3.1*, these univariate point sets for level-2 and level-3 should match univariate polynomials up to the 3rd order and the 5th order, respectively.

Note that the above set of notations for the univariate quadrature points and weights is different from the notations used in the general univariate moment matching Eqs. (3.35)-(3.36), because the SGQ involves different levels of univariate quadradure points and weights and is difficult to adopt a uniform set of notations for all levels. From Eq. (3.38), for level-2, $\tilde{p}_2 = \hat{p}_1, \tilde{w}_1 = \hat{w}_1$, and $\tilde{w}_2 = \hat{w}_2$, the following equations should be satisfied.

$$\begin{cases} \hat{w}_1 + 2\hat{w}_2 = M_0 = 1 \\ 2\hat{w}_2\hat{p}_1^2 = M_2 = 1 \end{cases} \tag{3.39}$$

Solving these two equations for $\hat{w}_1$ and $\hat{w}_2$ leads to

$$\begin{cases} \hat{w}_1 = 1 - 1/\hat{p}_1^2 \\ \hat{w}_2 = 1/(2\hat{p}_1^2) \end{cases} \tag{3.40}$$

If the point set $\{-\hat{p}_1, 0, \hat{p}_1\}$ is the set of univariate GHQ points, $\hat{p}_1 = \sqrt{3}$, $\hat{w}_1 = \frac{2}{3}$, $\hat{w}_2 = \frac{1}{6}$ (see Section 2.2.1.5).

Similarly, for level-3, $\tilde{p}_2 = \hat{p}_2$, $\tilde{p}_3 = \hat{p}_3$, $\tilde{w}_1 = \hat{w}_3$, $\tilde{w}_2 = \hat{w}_4$ and $\tilde{w}_3 = \hat{w}_5$, the following equations should be satisfied

$$\begin{cases} \hat{w}_3 + 2\hat{w}_4 + 2\hat{w}_5 = M_0 = 1 \\ 2\hat{w}_4\hat{p}_2^2 + 2\hat{w}_5\hat{p}_3^2 = M_2 = 1 \\ 2\hat{w}_4\hat{p}_2^4 + 2\hat{w}_5\hat{p}_3^4 = M_4 = 3 \end{cases} \tag{3.41}$$

Solving these equations (if $\hat{p}_3 \neq \hat{p}_2$) for $\hat{w}_3$, $\hat{w}_4$, $\hat{w}_5$ yields

$$\begin{cases} \hat{w}_3 = 1 - 2\hat{w}_4 - 2\hat{w}_5 \\ \hat{w}_4 = (3 - \hat{p}_3^2)/[2\hat{p}_2^2(\hat{p}_2^2 - \hat{p}_3^2)] \\ \hat{w}_5 = (3 - \hat{p}_2^2)/[2\hat{p}_3^2(\hat{p}_3^2 - \hat{p}_2^2)] \end{cases} \tag{3.42}$$

If $\hat{p}_3 = \hat{p}_2$, for Eq. (3.41) to remain valid, we must have $\hat{p}_3 = \hat{p}_2 = \sqrt{3}$. Then, $\hat{w}_3 = \frac{2}{3}$, $\hat{w}_4 + \hat{w}_5 = \frac{1}{6}$.

Note that the points $\hat{p}_1$, $\hat{p}_2$ and $\hat{p}_3$ are tunable parameters. When the locations of these points are given, the weights can be determined from Eqs. (3.39) and (3.41).

Remark 3.4: The moment matching method has been used to analyze [36] or extend the UT [88]. The main difference of this section"s method from the methods used in those references is that only univariate moment matching is needed in this section and then the resultant univariate point sets and weights are extended to the *n*-dimensional point set and weights using the sparse-grid method. Thus, the computations of the points and weights are greatly simplified.

Remark 3.5: Other univariate point-selection methods can be used if the condition of *Theorem 3.1* is satisfied. For example, we can use 4*l*-1 symmetric points for the univariate quadrature point set with the accuracy level-*l*, which can match the $(4l\text{-}2)^{th}$ order moments. Increasing the number of univariate quadrature points increases the accuracy level of the univariate integral approximation. But it may not be sufficient to

increase the accuracy level of the *n*-dimensional SGQ. It is worth noting that the univariate I_l with accuracy level-*l* requires the minimum number of *l* univariate quadrature points. In fact, one can choose *l* symmetric univariate GHQ points and weights as the univariate quadrature points and weights to satisfy the level-*l* requirement. Therefore, the univariate GHQ is a subset of the univariate quadrature proposed in this section. The sparse Gauss-Hermite Quadrature [25] for multi-dimensional problems is a subset of the SGQ, since they are based on the same sparse-grid method.

3.2.2 Multi-dimensional Quadrature Points and Weights

The *n*-dimensional SGQ points and weights can be obtained by the sparse-grid method, i.e. Eq. (3.10). More details can be found in Section 3.1.1.

In the following, three typical cases of SGQ are deduced:

Case 1: $\hat{p}_1 = \hat{p}_2 = \hat{p}_3$

In this case, the point set $\{-\hat{p}_1, 0, \hat{p}_1\}$ with the weights $(\hat{w}_2, \hat{w}_1, \hat{w}_2)$ is used for both level-2 and level-3 SGQ, and there are three variables, $\hat{p}_1$, $\hat{w}_1$ and $\hat{w}_2$. By Eqs. (3.39) and (3.41), we find that $\hat{p}_1 = \sqrt{3}$, $\hat{w}_1 = \frac{2}{3}$, and $\hat{w}_2 = \frac{1}{6}$. Note that this set of univariate quadrature points and weights are the same as the set of univariate GHQ points and weights $(m_L = 3)$.

Since *L*=3, by Equation (3.10), q can be 0, 1, or 2. When $q = 0$, by Eq. (3.11), $\mathbf{N}_0^n = [1, \cdots, 1]^T$. This leads to one point $[0, \cdots, 0]^T$ with the weight $(-1)^2 \mathbf{C}_{n-1}^2 = (n-1)(n-2)/2$ calculated by Eq. (3.13).

When $q = 1$, one element of Ξ in $\mathbf{N}_1^n$ is 2 and others are 1. This leads to points $P_1^{(i)}$ with the weight $(-1)^1 \mathbf{C}_{n-1}^1 \left(\frac{1}{6} \times 1\right)$ calculated by Eq. (3.13), and n duplicated points $[0, \cdots, 0]^T$ with weight $(-1)^1 \mathbf{C}_{n-1}^1 \left(\frac{2}{3} \times 1\right)$ calculated by Eq. (3.13).

$$P_1^{(i)} = \begin{cases} \sqrt{3}\mathbf{e}_i & i = 1, \cdots 2n \\ -\sqrt{3}\mathbf{e}_{i-n} \end{cases} \tag{3.43}$$

When $q = 2$, the elements of Ξ in $\mathbf{N}_2^n$ have two possible combinations. For the first combination, one element of Ξ is 3 and the others are 1. For the second combination, two elements of Ξ are 2 and the others are 1. The first combination leads to points $P_1^{(i)}$ with weight $1/6$, and n duplicated points $[0, \cdots, 0]^T$ with weight $2/3$. Similarly, the second combination leads to points $P_2^{(i)}$ with weight $A_2 = (1/6) \times (1/6) = 1/36$, $n-1$ duplicated points $P_1^{(i)}$ with weight $(2/3) \times (1/6) = 2/18$, and $n(n-1)/2$ duplicated points $[0, \cdots, 0]^T$ with weight $(2/3) \times (2/3) = 4/9$.

$$P_2^{(i)} = \begin{cases} \sqrt{3}(\mathbf{e}_j + \mathbf{e}_k) & j,k = 1, \cdots n; j < k \\ \sqrt{3}(\mathbf{e}_j - \mathbf{e}_k) & j,k = 1, \cdots n; j < k \\ \sqrt{3}(-\mathbf{e}_j + \mathbf{e}_k) & j,k = 1, \cdots n; j < k \\ \sqrt{3}(-\mathbf{e}_j - \mathbf{e}_k) & j,k = 1, \cdots n; j < k \end{cases} \tag{3.44}$$

Since the point $[0, \cdots, 0]^T$ is duplicated in all cases, by Eq. (3.14), the final weight of point $[0, \cdots, 0]^T$, denoted by A_0, can be calculated by Eq. (3.4) as

$$\begin{aligned} A_0 &= (n-1)(n-2)/2 - 2n(n-1)/3 + 2n/3 + 4n(n-1)/18 \\ &= n^2/18 - 7n/18 + 1 \end{aligned} \tag{3.45}$$

Similarly, the final weight of point $P_1^{(i)}$ denoted by A_1 can be calculated by

$$\begin{aligned} A_1 &= -(n-1)/6 + 1/6 + 2(n-1)/18 \\ &= -n/18 + 2/9 \end{aligned} \tag{3.46}$$

Hence, $I_{n,3}(\boldsymbol{f})$ can be obtained.

$$I_{n,3}(\boldsymbol{f}) = A_0 \boldsymbol{f}(\mathbf{0}) + A_1 \sum_{i=1}^{2n} \boldsymbol{f}\left(P_1^{(i)}\right) + A_2 \sum_{i=1}^{n(n-1)/2} \boldsymbol{f}\left(P_2^{(i)}\right) \tag{3.47}$$

Case 2: $\hat{p}_1 = \hat{p}_2 \neq \hat{p}_3$

In this case, the point set $\{-\hat{p}_1, 0, \hat{p}_1\}$ with the weights $(\hat{w}_2, \hat{w}_1, \hat{w}_2)$ is used for level-2 and the point set $\{-\hat{p}_3, -\hat{p}_1, 0, \hat{p}_1, \hat{p}_3\}$ with the weights $(\hat{w}_5, \hat{w}_4, \hat{w}_3, \hat{w}_4, \hat{w}_5)$ is used for level-3. There are seven variables, $\hat{p}_1$, $\hat{p}_3$, $\hat{w}_1$, $\hat{w}_2$, $\hat{w}_3$, $\hat{w}_4$, and $\hat{w}_5$, where $\hat{p}_1$ and $\hat{p}_3$ are two tunable parameters. The weights $\hat{w}_i (i = 1, \cdots 5)$ are calculated from Eqs. (3.39) and (3.41).

When $q = 0$, a similar conclusion as for Case 1 can be obtained.

When $q = 1$, one element of Ξ in $\mathbf{N}_1^n$ is 2 and others are 1. This leads to points $P_1^{(i)}$ with the weight $(-1)^1 \mathbf{C}_{n-1}^1 (\hat{w}_2 \times 1)$ calculated by Eq. (3.10), and n duplicated points $[0, \cdots, 0]^T$ with the weight $(-1)^1 \mathbf{C}_{n-1}^1 (\hat{w}_1 \times 1)$ calculated by Eq. (3.10).

$$P_1^{(i)} = \begin{cases} \hat{p}_1 \mathbf{e}_i & i = 1, \cdots 2n \\ -\hat{p}_1 \mathbf{e}_{i-n} \end{cases} \tag{3.48}$$

When $q = 2$, the elements of Ξ in $\mathbf{N}_2^n$ have two possible combinations. For the first combination, one element of Ξ is 3 and the others are 1. For the second combination, two elements of Ξ are 2, and the others are 1. The first combination leads to points $P_1^{(i)}$ with weight $\hat{w}_4$, $P_2^{(i)}$ with weight $A_2 = \hat{w}_5$, and n duplicated points $[0, \cdots, 0]^T$ with weight $\hat{w}_3$. Similarly, the second combination leads to points $P_3^{(i)}$ with weight $A_3 = \hat{w}_2^2$, $n-1$ duplicated $P_1^{(i)}$ with weight $\hat{w}_1 \hat{w}_2$, and $n(n-1)/2$ duplicated $[0, \cdots, 0]^T$ with weight $\hat{w}_1^2$.

$$P_3^{(i)} = \begin{cases} \hat{p}_1 (\mathbf{e}_j + \mathbf{e}_k) & j, k = 1, \cdots n; j < k \\ \hat{p}_1 (\mathbf{e}_j - \mathbf{e}_k) & j, k = 1, \cdots n; j < k \\ \hat{p}_1 (-\mathbf{e}_j + \mathbf{e}_k) & j, k = 1, \cdots n; j < k \\ \hat{p}_1 (-\mathbf{e}_j - \mathbf{e}_k) & j, k = 1, \cdots n; j < k \end{cases} \tag{3.49}$$

Since the point $[0,\cdots,0]^T$ is duplicated in all cases, by Eq. (3.14), the final weight of point $[0,\cdots,0]^T$, denoted by A_0, can be calculated by Eq. (3.14) as:

$$A_0 = (n-1)(n-2)/2 - n(n-1)\hat{w}_1 + m\hat{w}_3 + n(n-1)w_1^2/2 \tag{3.50}$$

Similarly, the final weight of point $P_1^{(i)}$, denoted by A_1, can be calculated by

$$A_1 = -(n-1)\hat{w}_2 + (n-1)\hat{w}_1\hat{w}_2 + \hat{w}_4 \tag{3.51}$$

Hence, $I_{n,3}(\boldsymbol{f})$ can be obtained.

$$I_{n,3}(\boldsymbol{f}) = A_0\boldsymbol{f}(\mathbf{0}) + A_1\sum_{i=1}^{2n}\boldsymbol{f}\left(P_1^{(i)}\right) + A_2\sum_{i=1}^{2n}\boldsymbol{f}\left(P_2^{(i)}\right) + A_3\sum_{i=1}^{n(n-1)/2}\boldsymbol{f}\left(P_3^{(i)}\right) \tag{3.52}$$

Case 1 is a special case of Case 2. If $\hat{p}_1 = \hat{p}_2 = \hat{p}_3 = \sqrt{3}$, one can get $\hat{w}_1 = \hat{w}_3 = \frac{2}{3}$, $\hat{w}_2 = 2\hat{w}_4 = 2\hat{w}_5 = \frac{1}{6}$. Additionally $[0,\cdots,\pm\hat{p}_1,\cdots,0]^T$ and $[0,\cdots,\pm\hat{p}_3,\cdots,0]^T$ become identical and the weights become $0.5(n-1)\left(n-2+m\hat{w}_1^2\right) - n(n-1)\hat{w}_1 + m\hat{w}_3 = \frac{n^2}{18} - \frac{7n}{18} + 1$, $(n-1)\hat{w}_2(\hat{w}_1 - 1) + \hat{w}_4 + \hat{w}_5 = -\frac{n}{18} + \frac{2}{9}$, and $\hat{w}_2^2 = \frac{1}{36}$, which are the same weights as those in Case 1.

Case 3: $\hat{p}_1 \neq \hat{p}_2 \neq \hat{p}_3$

In this case, the point set $\{-\hat{p}_1, 0, \hat{p}_1\}$ with the weights $(\hat{w}_2, \hat{w}_1, \hat{w}_2)$ is used for level-2 and the point set $\{-\hat{p}_3, -\hat{p}_2, 0, \hat{p}_2, \hat{p}_3\}$ with the weights $(\hat{w}_5, \hat{w}_4, \hat{w}_3, \hat{w}_4, \hat{w}_5)$ is used for level-3. There are eight variables, $\hat{p}_1$, $\hat{p}_2$, $\hat{p}_3$, $\hat{w}_1$, $\hat{w}_2$, $\hat{w}_3$, $\hat{w}_4$, and $\hat{w}_5$, where $\hat{p}_1$, $\hat{p}_2$, $\hat{p}_3$ are three tunable parameters. The weights $\hat{w}_i (i = 1,\cdots 5)$ are calculated from Eqs. (3.40) and (3.42).

When $q = 0$ and $q = 1$, similar conclusions as for Case 1 and Case 2 can be obtained.

When $q = 2$, the elements of Ξ in $\mathbf{N}_2^n$ have two possible combinations. For the first combination, one element of Ξ is 3 and others are 1. For the second combination,

two elements of Ξ are 2 and others are 1. The first combination leads to points $P_2^{(i)}$ with weight $A_2 = \hat{w}_4$, $P_3^{(i)}$ with weight $A_3 = \hat{w}_5$, and n duplicated points $[0,\cdots,0]^T$ with weight $\hat{w}_3$. Similarly, the second combination leads to points $P_4^{(i)}$ with weight $A_4 = \hat{w}_2^2$, $n-1$ duplicated $P_1^{(i)}$ with weight $\hat{w}_1\hat{w}_2$, and $n(n-1)/2$ duplicated $[0,\cdots,0]^T$ with weight $\hat{w}_1^2$.

$$P_4^{(i)} = \begin{cases} \hat{p}_1(\mathbf{e}_j + \mathbf{e}_k) & j,k = 1,\cdots n; j < k \\ \hat{p}_1(\mathbf{e}_j - \mathbf{e}_k) & j,k = 1,\cdots n; j < k \\ \hat{p}_1(-\mathbf{e}_j + \mathbf{e}_k) & j,k = 1,\cdots n; j < k \\ \hat{p}_1(-\mathbf{e}_j - \mathbf{e}_k) & j,k = 1,\cdots n; j < k \end{cases} \tag{3.53}$$

Since the point $[0,\cdots,0]^T$ is duplicated in all cases, the final weight of point $[0,\cdots,0]^T$, denoted by A_0, can be calculated by Eq. (3.10) or Eq. (3.14) as:

$$A_0 = (n-1)(n-2)/2 - n(n-1)\hat{w}_1 + m\hat{w}_3 + n(n-1)\hat{w}_1^2/2 \tag{3.54}$$

Similarly, the final weight of point $P_1^{(i)}$, denoted by A_1, can be calculated by

$$A_1 = -(n-1)\hat{w}_2 + (n-1)\hat{w}_1\hat{w}_2 \tag{3.55}$$

Hence, $I_{n,3}(\boldsymbol{f})$ can be obtained.

$$I_{n,3}(\boldsymbol{f}) = A_0\boldsymbol{f}(\mathbf{0}) + A_1\sum_{i=1}^{2n}\boldsymbol{f}(P_1^{(i)}) + A_2\sum_{i=1}^{2n}\boldsymbol{f}(P_2^{(i)}) + A_3\sum_{i=1}^{2n}\boldsymbol{f}(P_3^{(i)}) + A_4\sum_{i=1}^{n(n-1)/2}\boldsymbol{f}(P_4^{(i)}) \tag{3.56}$$

Note that Case 2 is a special case of Case 3. If $\hat{p}_1 = \hat{p}_2$, then $[0,\cdots,\pm\hat{p}_1,\cdots,0]^T$ and $[0,\cdots,\pm\hat{p}_2,\cdots,0]^T$ become identical, and the weights for these redundant points should be re-calculated as $(n-1)\hat{w}_2(\hat{w}_1 - 1) + \hat{w}_4$, which is the same as the corresponding weights in Case 2.

3.2.3 Comparison of the sparse-grid quadrature rules, the unscented transformation, and the Gauss-Hermite quadrature rule

The following theorem shows that the sigma-points and weights generated by the UT is a subset of the SGQ.

Theorem 3.3: The points and weights generated by the UT are identical to the points and weights generated by the SGQ rule with level-2 accuracy if one point and three symmetric points are used for the level-1 and level-2 univariate quadrature point set, respectively.

Proof: For the level-1 univariate quadrature rule I_1, the point set is $\{0\}$ with the corresponding weight of 1. For the level-2 univariate quadrature I_2, the univariate quadrature point set is $\{-\hat{p}_1, 0, \hat{p}_1\}$ with the corresponding weight sequence $(\hat{w}_2, \hat{w}_1, \hat{w}_2)$.

Since the accuracy level of the SGQ is *L*=2, the value of q in $\mathbf{N}_q^n$ can be 0 or 1. When $q=0$, $\mathbf{N}_0^n = \left\{ \underbrace{(1,1,\cdots,1,1)}_{n \text{ elements}} \right\}$, the *n*-dimensional SGQ point corresponding to $\mathbf{N}_0^n$ will be $[0,0,\cdots 0,0]^T$ with the corresponding weight of $(-1)^{2-1-0} \times \mathbf{C}_{n-1}^{2-1-0} \times \underbrace{\left(\overbrace{1\times 1\times\cdots\times 1\times 1}^{n-1 \text{ elements}} \right)}_{\text{product of weights}} = -(n-1)$ (using Eq. (3.13)). When $q=1$, the set of accuracy level sequences becomes $\mathbf{N}_1^n = \underbrace{\left\{ \overbrace{(2,1,\cdots 1,1)}^{n}, \overbrace{(1,2,\cdots 1,1)}^{n}, \cdots \overbrace{(1,\cdots 1,2,1,\cdots 1)}^{n}, \cdots \overbrace{(1,1,\cdots 1,2)}^{n} \right\}}_{n \text{ elements}}$. Corresponding to the sequence $(2,1,\cdots 1,1)$, there are three *n*-dimensional SGQ points $[-\hat{p}_1, 0, \cdots 0, 0]^T$, $[0,0,\cdots 0,0]^T$, and $[\hat{p}_1, 0, \cdots 0, 0]^T$. The weights corresponding to $[-\hat{p}_1, 0, \cdots 0, 0]^T$ and $[\hat{p}_1, 0, \cdots 0, 0]^T$ are the same, i.e. $W_2 = (-1)^{2-1-1} \times \mathbf{C}_{n-1}^{2-1-1} \times \underbrace{\left(\overbrace{1\times 1\times\cdots\times 1}^{n-1 \text{ elements}} \times \hat{w}_2 \right)}_{\text{product of weights}} = \hat{w}_2$ (using Eq. (3.13)). Moreover, for this sequence, the SGQ can match the polynomials of the form $\left(ax_1^3 + bx_1^2 + cx_1 + d\right) x_2^{k_1} x_3^{k_2} \cdots x_n^{k_n}$, where a, b, c, and d are real numbers, and $k_1, \cdots, k_n$ can be 0 or 1, because I_2 is exact for all univariate polynomials of the order up to

$2\times 2-1=3$ and I_1 is exact for all univariate polynomials of the order up to $2\times 1-1=1$, according to *Theorem 3.1*. Similarly, for other sequences in $\mathbf{N}_1^n$ such as $(1,\cdots,1,2,1,\cdots 1)$, there are three *n*-dimensional SGQ points $[0,\cdots,0,-\hat{p}_1,0,\cdots 0]^T$, $[0,\cdots,0,0,0,\cdots 0]^T$, and $[0,\cdots,0,\hat{p}_1,0,\cdots 0]^T$. The weights corresponding to $[0,\cdots,0,-\hat{p}_1,0,\cdots 0]^T$ and $[0,\cdots,0,\hat{p}_1,0,\cdots 0]^T$ are the same, i.e. $W_2=\hat{w}_2$ (using Eq. (3.13)). There are *n* such combinations of accuracy level sequences in $\mathbf{N}_1^n$ with the similar calculation of *n*-dimensional SGQ points and weights. Note that since the point $[0,\cdots,0,0,0,\cdots 0]^T$ is a repeated point appearing once in $\mathbf{N}_0^n$ and *n* times in $\mathbf{N}_1^n$, Eq. (3.14) is used to calculate its

$$W_1=(-1)^{2-1-0}\times \mathbf{C}_{n-1}^{2-1-0}\times \underbrace{\left(\overbrace{1\times 1\times\cdots\times 1\times 1}^{n \text{ elements}}\right)}_{\text{product of weights}}+(-1)^{2-1-1}\times \mathbf{C}_{n-1}^{2-1-1}$$

weight as $\times\left\{\overbrace{\underbrace{\left(\overbrace{\hat{w}_1\times 1\times 1\times\cdots\times 1}^{n-1 \text{ elements}}\right)}_{\text{product of weights}}+\underbrace{\left(\overbrace{1\times\hat{w}_1\times 1\times 1\times\cdots\times 1}^{n-2 \text{ elements}}\right)}_{\text{product of weights}}+\cdots+\underbrace{\left(\overbrace{1\times 1\times\cdots\times 1\times\hat{w}_1}^{n-1 \text{ elements}}\right)}_{\text{product of weights}}}^{n \text{ elements}}\right\}$.

$$=-(n-1)+n\cdot\hat{w}_1$$

To summarize, the points and weights of the SGQ with level-2 accuracy are

$$\boldsymbol{\gamma}_i=\begin{cases}[0,0,\cdots 0,0]^T; & i=1\\ \hat{p}_1\mathbf{e}_{i-1}; & 2\le i\le n+1\\ -\hat{p}_1\mathbf{e}_{i-n-1}; & n+2\le i\le 2n+1\end{cases} \tag{3.57}$$

and

$$W_i=\begin{cases}-(n-1)+n\cdot\hat{w}_1 & i=1\\ \hat{w}_2 & i=2\cdots 2n+1\end{cases} \tag{3.58}$$

respectively. If we choose $\hat{p}_1=\sqrt{n+\kappa}$, by Eq. (3.40), we obtain $\hat{w}_1=1-\frac{1}{(n+\kappa)}$ and $\hat{w}_2=\frac{1}{2(n+\kappa)}$. Hence,

$$W_i = \begin{cases} -(n-1)+n\cdot\left(1-\dfrac{1}{(n+\kappa)}\right)=\dfrac{\kappa}{n+\kappa} & i=1 \\ \dfrac{1}{2(n+\kappa)} & i=2\cdots 2n+1 \end{cases} \tag{3.59}$$

Comparing Eqs. (3.57) and (3.59) with the sigma points and weights of the UT in Eq. (2.46), they are identical. ■

Theorem 3.3 implies that the UKF is a subset of the SGQF at the accuracy level-2. The SGQF can achieve higher accuracy than the UKF by increasing L.

Proposition 3.2: If $X_1 \subset X_2, 1<q<n$, the SGQ points corresponding to accuracy level set $\mathbf{N}_{q-1}^n$ are contained in the SGQ point set corresponding to $\mathbf{N}_q^n$.

Proof: By the definition, $\mathbf{N}_{q-1}^n = \left\{\Xi : \sum_{j=1}^{n} i_j = n+q-1\right\}$ and $\mathbf{N}_q^n = \left\{\Xi : \sum_{j=1}^{n} i_j = n+q-1+1\right\}$. If the sequence $(i_1, i_2, \cdots, i_j, \cdots, i_n)$ belongs to $\mathbf{N}_{q-1}^n$, then the sequence $(i_1, i_2, \cdots, i_j+1, \cdots i_n)$ belongs to $\mathbf{N}_q^n$. For $\mathbf{N}_q^n$, each element sequence in it contains at most q elements that are greater than 1, which has been discussed in the proof of *Proposition 3.1*. Because $1<q<n$, and there are at most q elements greater than 1 in the accuracy level sequence, we can always find at least one $i_j = 1,\ 1 \le j \le n$ such that the points corresponding to $(i_1, i_2, \cdots, i_j, \cdots, i_n)$ are generated by the tensor product rule $X_{i_1} \otimes X_{i_2} \otimes \cdots X_1 \cdots \otimes X_{i_n}$, whereas the points corresponding to $(i_1, i_2, \cdots, i_j+1, \cdots, i_n)$ are generated by the tensor product rule $X_{i_1} \otimes X_{i_2} \otimes \cdots X_2 \cdots \otimes X_{i_n}$. Because $X_1 \subset X_2$, any point generated by $\mathbf{N}_{q-1}^n$ will be contained in $\mathbf{N}_q^n$. ■

There are several important advantages of the SGQ over other point-based methods.

(1) SGQ with different accuracy levels can be implemented using the same sparse-grid framework. When univariate quadrature points and weights of different accuracy levels are given, the multidimensional SGQ point set and weights for a given

accuracy level can be uniquely determined. Theoretically, if the nonlinear function $\boldsymbol{f}$ can be approximated sufficiently well by polynomials, then the integral $\int_{-\infty}^{\infty} \boldsymbol{f}(\mathbf{x})\mathbf{N}(\mathbf{x};\mathbf{0},\mathbf{I})d\mathbf{x}$ can be calculated with sufficient accuracy by increasing the level of SGQ. Other point-based methods are difficult to extend to achieve higher level accuracy. For example, the integral $\int_{-\infty}^{\infty}\left(1+\sum x_i^2\right)^n \mathbf{N}(\mathbf{x};\mathbf{0},\mathbf{I})d\mathbf{x}$ (x_i is the i^{th} element of the vector $\mathbf{x}$) can be calculated exactly by the SGQ with accuracy level-4 and level-5 when n equals 3 or 4, whereas with UT or cubature rule, it is difficult to achieve this accuracy.

(2) The number of SGQ points for a fixed accuracy level increases polynomially with dimension. The classical method to extend the one-dimensional integral approximation rule to the multi-dimensional problem is the direct tensor product rule. The multi-dimensional GHQ rule uses this method and the number of the multi-dimensional GHQ points increases exponentially with dimension, which severely limits the applicability of this rule. Using the SGQ points based on the sparse-grid method, however, the SGQF algorithm needs far fewer points. Thus, it alleviates the computational load problem, and is very efficient for high dimensional estimation problems.

(3) The tunable parameters $\hat{p}_1, \hat{p}_2, \hat{p}_3, \cdots$ make the SGQF more flexible than the GHQF or the SGHQF. Since the point-based methods evaluate the value of the nonlinear function at those points, it is understandable that the location of those points may affect the accuracy of the integral result. For different nonlinear functions, different point-sets with the same accuracy level may have very different performance. These tunable parameters can capture non-Gaussian pdfs to some extent. It is similar to the UKF, which has an adjustable parameter κ to tune the performance depending on applications. In comparison with the GHQ rule, the locations of the GHQ points are fixed, which is not

flexible for non-Gaussian problems. Instead of using fixed points, the SGQ can adjust the locations of the points. When the univariate GHQ rule is used as the univariate quadrature rule for the SGQ, SGQ will have the same points and weights as the Sparse Gauss Hermite Quadrature (SGHQ) rule. In contrast to the SGHQ, the SGQ has tunable parameters. When selecting univariate quadrature points, the univariate GHQ points can be used for the SGQ as the reference values of the tunable parameters. Moreover, it is worth noting that the SGQ can use nested sequences of quadrature rules, which means that the univariate quadrature point set with the lower accuracy level is a subset of the univariate quadrature point set with the higher accuracy level, i.e. $X_i \subset X_j$ if $i < j$. For example, in Case 2 of the level-3 SGQ, the univariate quadrature point set $\{-\hat{p}_1, 0, \hat{p}_1\}$ at level-2 is nested in the univariate quadrature point set $\{-\hat{p}_3, -\hat{p}_2, 0, \hat{p}_2, \hat{p}_3\}$ at level-3 because $\hat{p}_1 = \hat{p}_2$. The nested structure can reduce the total number of the SGQ points when extended to the multi-dimensional point set using the sparse-grid method, compared with other non-nested SGQs like SGHQ.

3.2.4 The relationship between the sparse-grid quadrautre filters and the cubature Kalman filters

By Eq. (2.61), when $\boldsymbol{g}(\mathbf{x}) = \mathbf{x}^{|\alpha|} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, $|\alpha| = \alpha_1 + \cdots \alpha_n$, $\boldsymbol{g}(r\mathbf{s}) = r^{|\alpha|}\mathbf{s}^{|\alpha|}$,

$$\begin{aligned} I\left(\mathbf{x}^{|\alpha|}\right) &= \int_{\mathbb{R}^n} x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \exp\left(-\mathbf{x}^T\mathbf{x}\right) d\mathbf{x} \\ &= \int_0^\infty \int_{U_n} (rs_1)^{\alpha_1} (rs_2)^{\alpha_2} \cdots (rs_n)^{\alpha_n} r^{n-1} \exp\left(-r^2\right) d\sigma(\mathbf{s}) dr \\ &= \int_0^\infty r^{\alpha_1+\alpha_2+\cdots+\alpha_n} r^{n-1} \exp\left(-r^2\right) dr \int_{U_n} s_1^{\alpha_1} s_2^{\alpha_2} \cdots s_n^{\alpha_n} d\sigma(\mathbf{s}) \\ &= \int_0^\infty r^{n-1+|\alpha|} \exp\left(-r^2\right) dr \int_{U_n} \mathbf{s}^{|\alpha|} d\sigma(\mathbf{s}) \end{aligned} \tag{3.60}$$

Equation (3.60) can be rewritten as

$$\int_{U_n} \mathbf{s}^{|\alpha|} d\sigma(\mathbf{s}) = I\left(\mathbf{x}^{|\alpha|}\right) \Big/ \int_0^\infty r^{n-1+|\alpha|} \exp\left(-r^2\right) dr \tag{3.61}$$

Definition 3.1: The integral $\int_{\mathbb{R}^n} w_g(\mathbf{x})\boldsymbol{g}(\mathbf{x})d\mathbf{x} \simeq \sum_i W_i \boldsymbol{g}(\boldsymbol{\gamma}_i)$ ($\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n]^T \in \mathbb{R}^n$, and $w_g(\mathbf{x})$ is a given weight function) is a d^{th}-degree rule if it is exact for a $\boldsymbol{g}(\mathbf{x})$ that has the form of a linear combination of monomials $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ with the total order up to d ($\alpha_1, \alpha_2, \cdots, \alpha_n$ are nonnegative integers and $0 \le \alpha_1 + \alpha_2 + \cdots + \alpha_n \le d$) and there is at least one monomial of degree d+1 for which this rule is not exact [58].■

Note that if $I(\boldsymbol{g}) = \sum_{j=1}^{N_p} \bar{W}_j \boldsymbol{g}(\bar{\boldsymbol{\gamma}}_j)$ is a d^{th}-degree quadrature rule, then the spherical rules $\int_{U_n} \boldsymbol{g}(\mathbf{s}) d\sigma(\mathbf{s})$ has d^{th}-degree accuracy. The sparse-grid quadrature rule with accuracy level-L is a $(2L\text{-}1)^{\text{th}}$-degree rule.

The sparse-grid method can be used to obtain the spherical rule with different accuracies by Eq. (3.61). Because the points $\mathbf{s}_j$ of the spherical rules all satisfy the constraint $\|\mathbf{s}_j\|_2 = 1$, the points of the spherical rule can be obtained by projecting the sparse-grid quadrature points $\bar{\boldsymbol{\gamma}}_j$ in Euclidean space onto the hyper-sphere, i.e.

$$\mathbf{s}_j = \frac{\bar{\boldsymbol{\gamma}}_j}{\|\bar{\boldsymbol{\gamma}}_j\|_2} \tag{3.62}$$

where $\|\cdot\|_2$ is the norm operation. Note that the origin point is omitted when it is projected onto the hyper-sphere because the origin point does not have influence on the accuracy analysis.

The equation (3.61) can be rewritten by

$$\begin{aligned}
&\int_{U_n} \mathbf{s}^{|\alpha|} d\sigma(\mathbf{s}) = I\left(\mathbf{x}^{|\alpha|}\right) \Big/ \int_0^\infty r^{n-1+|\alpha|} \exp\left(-r^2\right) dr \\
&= \sum_{j=1}^{N_p} \bar{W}_j \left(\bar{\boldsymbol{\gamma}}_j\right)^{|\alpha|} \Big/ \int_0^\infty r^{n-1+|\alpha|} \exp\left(-r^2\right) dr \\
&= \sum_{j=1}^{N_p} \bar{W}_j \left(\mathbf{s}_j \cdot \|\bar{\boldsymbol{\gamma}}_j\|_2\right)^{|\alpha|} \Big/ \int_0^\infty r^{n-1+|\alpha|} \exp\left(-r^2\right) dr \\
&= \sum_{j=1}^{N_p} \bar{W}_j \left(\mathbf{s}_j\right)^{|\alpha|} \|\bar{\boldsymbol{\gamma}}_j\|_2^{|\alpha|} \Big/ \int_0^\infty r^{n-1+|\alpha|} \exp\left(-r^2\right) dr
\end{aligned} \tag{3.63}$$

where N_p is the number of sparse-grid quadrature points.

Remark 3.6 It is not necessary to use Eq. (3.63) when $|\alpha|$ is an odd number, because both the value of $I\left(\mathbf{x}^{|\alpha|}\right)$ and $\int_{U_n}\mathbf{s}^{|\alpha|}d\sigma(\mathbf{s})$ are 0.

By Eq. (3.63), the weights $w_{s,j}$ of the spherical rule are given by

$$w_{s,j}=\bar{W}_j\left\|\bar{\boldsymbol{\gamma}}_j\right\|_2^{|\alpha|}\Big/\int_0^{\infty}r^{n-1+|\alpha|}\exp\left(-r^2\right)dr=\bar{W}_j\left\|\bar{\boldsymbol{\gamma}}_j\right\|_2^{|\alpha|}\Big/\left(\frac{\Gamma\left(n/2+|\alpha|/2\right)}{2}\right) \tag{3.64}$$

If the spherical rule is obtained by the projection of the sparse-grid quadrature rule, a new cubature rule can be obtained by combining the new spherical rule and the radial rule in Eq. (2.66). This new cubature rule is referred to as the projected sparse-grid rule. The following theorem reveals the relationship between the projected sparse-grid rule and the third-degree cubature Kalman filter in [37].

Theorem 3.4: If the d^{th} degree (d is even) fully symmetric quadrature rule is used in Eq. (3.61), the induced spherical rule in Eq. (3.63) with the points given by Eq. (3.62) and the weights given by Eq. (3.64) is also a d^{th} degree rule.

Proof: The spherical rule in Eq. (3.63), which is induced from the d^{th} degree quadrature rule, is exact for $\int_{U_n}\mathbf{s}^d d\sigma(\mathbf{s})$. To show that it is a d^{th} degree spherical rule, we need to show that it is exact for any spherical rules $\int_{U_n}\mathbf{s}^\beta d\sigma(\mathbf{s})$ with degree $0\le\beta<d$.

$$\begin{aligned}\int_{U_n}\mathbf{s}^\beta d\sigma(\mathbf{s})&=\frac{\int_0^{\infty}r^d r^{n-1}\exp\left(-r^2\right)dr}{\int_0^{\infty}r^d r^{n-1}\exp\left(-r^2\right)dr}\int_{U_n}\mathbf{s}^\beta d\sigma(\mathbf{s})\\&=\frac{\int_0^{\infty}r^{d-\beta}r^\beta r^{n-1}\exp\left(-r^2\right)dr}{\int_0^{\infty}r^d r^{n-1}\exp\left(-r^2\right)dr}\int_{U_n}\mathbf{s}^\beta d\sigma(\mathbf{s})\\&=\frac{\int_0^{\infty}r^{d-\beta}r^\beta r^{n-1}\exp\left(-r^2\right)dr\int_{U_n}\mathbf{s}^\beta d\sigma(\mathbf{s})}{\left(\frac{\Gamma\left(n/2+d/2\right)}{2}\right)}\end{aligned} \tag{3.65}$$

Let $\mathbf{x} = r\mathbf{s}$ with $\mathbf{s}^T\mathbf{s} = 1$ and $r = \sqrt{\mathbf{x}^T\mathbf{x}}$. The numerator in the last equality of Eq. (3.65) can be rewritten as

$$\begin{aligned} &\int_0^\infty r^{d-\beta} r^\beta r^{n-1} \exp\left(-r^2\right) dr \int_{U_n} \mathbf{s}^\beta d\sigma(\mathbf{s}) \\ &= \int_{\mathbb{R}^n} \left[\left(\sqrt{x_1^2 + \cdots + x_n^2} \right)^{d-\beta} \mathbf{x}^\beta \right] \exp\left(-\mathbf{x}^T\mathbf{x}\right) d\mathbf{x} \end{aligned} \tag{3.66}$$

If β is odd, $\left[\left(\sqrt{x_1^2 + \cdots + x_n^2} \right)^{d-\beta} \mathbf{x}^\beta \right]$ is not a polynomial but still an odd function.

Because of the symmetry of the integration region and the property of the odd function, the integral in Eq. (3.66) vanishes,

$$\int_{\mathbb{R}^n} \left[\left(\sqrt{x_1^2 + \cdots + x_n^2} \right)^{d-\beta} \mathbf{x}^\beta \right] \exp\left(-\mathbf{x}^T\mathbf{x}\right) d\mathbf{x} = 0 \tag{3.67}$$

So, for any odd $\beta < d$,

$$\int_{U_n} \mathbf{s}^\beta d\sigma(\mathbf{s}) = 0 = \sum_{j=1}^{N_p} w_{s,j} \left(\mathbf{s}_j\right)^\beta \tag{3.68}$$

The second equality is true due to the fact that the spherical rule is fully symmetric. If β is even, then $d - \beta$ is even, $\left[\left(\sqrt{x_1^2 + \cdots + x_n^2} \right)^{d-\beta} \mathbf{x}^\beta \right]$ is a polynomial with degree $\frac{d-\beta}{2} + \beta \le \frac{d+\beta}{2} \le \frac{d+d}{2} = d$. Therefore, the integral in Eq. (3.66) can be exactly calculated by the d^{th} degree quadrature rule as follows

$$\begin{aligned} &\int_{\mathbb{R}^n} \left[\left(\sqrt{x_1^2 + \cdots + x_n^2} \right)^{d-\beta} \mathbf{x}^\beta \right] \exp\left(-\mathbf{x}^T\mathbf{x}\right) d\mathbf{x} \\ &= \sum_{j=1}^{N_p} \bar{W}_j \left\| \bar{\boldsymbol{\gamma}}_j \right\|_2^{d-\beta} \bar{\boldsymbol{\gamma}}_j^\beta \end{aligned} \tag{3.69}$$

Combining Eqs. (3.65), (3.66), and (3.69) leads to

$$\begin{aligned}\int_{U_n}\mathbf{s}^{\beta}d\sigma(\mathbf{s}) &= \sum_{j=1}^{N_p}\bar{W}_j\left\|\bar{\boldsymbol{\gamma}}_j\right\|_2^{d-\beta}\bar{\boldsymbol{\gamma}}_j^{\beta}\Bigg/\left(\frac{\Gamma\left(n/2+d/2\right)}{2}\right)\\ &= \sum_{j=1}^{N_p}\bar{W}_j\left\|\bar{\boldsymbol{\gamma}}_j\right\|_2^{d}\left(\frac{\bar{\boldsymbol{\gamma}}_j}{\left\|\bar{\boldsymbol{\gamma}}_j\right\|_2}\right)^{\beta}\Bigg/\left(\frac{\Gamma\left(n/2+d/2\right)}{2}\right)\\ &= \sum_{j=1}^{N_p}w_{s,j}\left(\mathbf{s}_j\right)^{\beta}\end{aligned} \tag{3.70}$$

where Eqs. (3.62) and (3.64) are used to arrive at the last equality. By Eqs. (3.68) and (3.70), it can be seen that any spherical rules with degree $\beta < d$ can be exactly calculated by Eq. (3.63). ■

Theorem 3.5: The third-degree cubature rule is a subset of the projected sparse-grid rule at the accuracy level-2.

Proof: It has been proven that the unscented transformation is a subset of the sparse-grid quadrature rules at the accuracy level-2 in Section 3.2. If we can prove that the third-degree cubature rule is identical to the projected unscented transformation, the theorem is equivalently proved.

The quadrature rule using unscented transformation is given by

$$\int_{\mathbb{R}^n}\boldsymbol{g}(\mathbf{x})N(\mathbf{x};\mathbf{0},\mathbf{I})d\mathbf{x}\approx\frac{\kappa}{(n+\kappa)}\boldsymbol{g}(\mathbf{0})+\frac{1}{2(n+\kappa)}\sum_{j=1}^{n}\left(\boldsymbol{g}\left(\sqrt{(n+\kappa)}\mathbf{e}_j\right)+\boldsymbol{g}\left(-\sqrt{(n+\kappa)}\mathbf{e}_j\right)\right) \tag{3.71}$$

where κ is the tunable parameter.

Hence,

$$I(\boldsymbol{g})=\sum_{j=1}^{2n+1}\bar{W}_i\boldsymbol{g}\left(\bar{\boldsymbol{\gamma}}_j\right)=\frac{\kappa\pi^{n/2}}{(n+\kappa)}\boldsymbol{g}(\mathbf{0})+\frac{\pi^{n/2}}{2(n+\kappa)}\sum_{j=1}^{n}\left(\boldsymbol{g}\left(\sqrt{\frac{(n+\kappa)}{2}}\mathbf{e}_j\right)+\boldsymbol{g}\left(-\sqrt{\frac{(n+\kappa)}{2}}\mathbf{e}_j\right)\right) \tag{3.72}$$

The points $\bar{\boldsymbol{\gamma}}_j$ and weights $\bar{W}_j$ are given by

$$\bar{\boldsymbol{\gamma}}_j = \begin{cases} \mathbf{0} & j = 1 \\ \sqrt{(n+\kappa)/2}\,\mathbf{e}_{j-1} & j = 2,\cdots,n+1 \\ -\sqrt{(n+\kappa)/2}\,\mathbf{e}_{j-n-1} & j = n+2,\cdots,2n+1 \end{cases} \tag{3.73}$$

and

$$\bar{W}_j = \begin{cases} \dfrac{\kappa}{(n+\kappa)}\pi^{n/2} & j = 1 \\ \dfrac{1}{2(n+\kappa)}\pi^{n/2} & j = 2,\cdots,2n+1 \end{cases} \tag{3.74}$$

Hence, the points $\mathbf{s}_j$ of the spherical rule are given by

$$\mathbf{s}_j = \frac{\bar{\boldsymbol{\gamma}}_j}{\left\|\bar{\boldsymbol{\gamma}}_j\right\|_2} = \begin{cases} \mathbf{e}_j & j = 1,\cdots,n \\ -\mathbf{e}_{j-n} & j = n+1,\cdots,2n \end{cases} \tag{3.75}$$

Note that the origin point is omitted.

By *Remark 3.6* and *Theorem 3.4*, for the third-degree rule, $|\alpha|$ is chosen to be 2.

The weights $w_{s,j}$ of the spherical rule can be obtained by

$$\begin{aligned} w_{s,j} &= \bar{W}_j \left\|\bar{\boldsymbol{\gamma}}_j\right\|_2^{|\alpha|} \Big/ \int_0^\infty r^{n-1+|\alpha|}\exp\left(-r^2\right)dr \\ &= \bar{W}_j \left\|\bar{\boldsymbol{\gamma}}_j\right\|_2^{|\alpha|} \Big/ \left(\frac{\Gamma\left(n/2+|\alpha|/2\right)}{2}\right) \\ &= \frac{1}{2(n+\kappa)}\pi^{n/2}\cdot\frac{(n+\kappa)}{2} \Big/ \left(\frac{n/2\,\Gamma(n/2)}{2}\right) \\ &= \pi^{n/2} \Big/ \left(n\Gamma\left(n/2\right)\right) = A_n/2n \end{aligned} \tag{3.76}$$

Hence, by Eqs. (3.63), (3.75), and (3.76), the third-degree spherical rule can be obtained

$$I_{U_n,3} = \frac{A_n}{2n}\sum_{i=1}^{n}\left(g\left(\mathbf{e}_i\right)+g\left(-\mathbf{e}_i\right)\right) \tag{3.77}$$

Equation (3.77) is identical to the third-degree spherical rule in Eq. (2.65).

Note that the third-degree radial rule can be obtained by the procedure shown in Section 2.2.1.6. Recall that the unscented transformation is a subset of the sparse-grid quadrature rules (level-2). Hence, the third-degree cubature rule [37] is a subset of the projected sparse-grid rule (level-2).■

Note that an arbitrarily accurate cubature rule can be obtained by combining the projection of the sparse-grid quadrature rule with the moment matching method. For example, a fifth-degree cubature rule can be obtained as follows.

One of the fifth-degree (accuracy level-3) sparse-grid quadrature rules can be described as in Section 3.2, Eq. (3.47)

$$\int_{\mathbb{R}^n} \boldsymbol{g}(\mathbf{x}) N(\mathbf{x};\mathbf{0},\mathbf{I}) d\mathbf{x} \approx A_0 \boldsymbol{g}(\mathbf{0}) + A_1 \sum_{j=1}^{2n} \boldsymbol{g}\left(P_j^{(1)}\right) + A_2 \sum_{j=1}^{n(n-1)/2} \boldsymbol{g}\left(P_j^{(2)}\right) \tag{3.78}$$

where $A_0 = n^2/18 - 7n/18 + 1$, $A_1 = -n/18 + 2/9$, $A_2 = 1/36$

and

$$P_j^{(1)} = \begin{cases} \sqrt{3}\mathbf{e}_j & j = 1,\cdots 2n \\ -\sqrt{3}\mathbf{e}_{j-n} \end{cases} \tag{3.79}$$

$$P_j^{(2)} = \begin{cases} \sqrt{3}\left(\mathbf{e}_i + \mathbf{e}_k\right) & i,k = 1,\cdots n; i < k \\ \sqrt{3}\left(\mathbf{e}_i - \mathbf{e}_k\right) & i,k = 1,\cdots n; i < k \\ \sqrt{3}\left(-\mathbf{e}_i + \mathbf{e}_k\right) & i,k = 1,\cdots n; i < k \\ \sqrt{3}\left(-\mathbf{e}_i - \mathbf{e}_k\right) & i,k = 1,\cdots n; i < k \end{cases} \tag{3.80}$$

We obtain the points $\overline{\boldsymbol{\gamma}}_j^{(1)}, \overline{\boldsymbol{\gamma}}_j^{(2)}$ and weights $\overline{W}_j^{(1)}$ and $\overline{W}_j^{(2)}$ by $\overline{\boldsymbol{\gamma}}_j^{(1)} = P_j^{(1)}/\sqrt{2}$, $\overline{\boldsymbol{\gamma}}_j^{(2)} = P_j^{(2)}/\sqrt{2}$, $\overline{W}_j^{(1)} = \overline{W}_j^{(1)} \cdot \pi^{n/2}$, and $\overline{W}_j^{(2)} = \overline{W}_j^{(2)} \cdot \pi^{n/2}$, respectively. Note that the origin point and weight are omitted.

The points $\mathbf{s}_j^{(1)}$ projected from $\overline{\boldsymbol{\gamma}}_j^{(1)}$ are given by

$$\mathbf{s}_j^{(1)} = \frac{\overline{\boldsymbol{\gamma}}_j}{\left\|\overline{\boldsymbol{\gamma}}_j\right\|_2} = \begin{cases} \mathbf{s}_j^{(1)^+} = \mathbf{e}_j & j = 1,\cdots,n \\ \mathbf{s}_j^{(1)^-} = -\mathbf{e}_{j-n} & j = n+1,\cdots,2n \end{cases} \tag{3.81}$$

The point sets $\left\{\mathbf{s}_j^{(2)}\right\}$ projected from $\bar{\boldsymbol{\gamma}}_j^{(2)}$ are given by

$$\mathbf{s}_j^{(2)} = \frac{\bar{\boldsymbol{\gamma}}_j}{\left\|\bar{\boldsymbol{\gamma}}_j\right\|_2} = \begin{cases} \mathbf{s}_{j1}^{(2)^+} = \sqrt{\frac{1}{2}}\left(\mathbf{e}_k + \mathbf{e}_l\right) & k < l, k, l = 1, 2, \cdots, n \\ \mathbf{s}_{j1}^{(2)^-} = -\sqrt{\frac{1}{2}}\left(\mathbf{e}_k + \mathbf{e}_l\right) & k < l, k, l = 1, 2, \cdots, n \\ \mathbf{s}_{j2}^{(2)^+} = \sqrt{\frac{1}{2}}\left(\mathbf{e}_k - \mathbf{e}_l\right) & k < l, k, l = 1, 2, \cdots, n \\ \mathbf{s}_{j2}^{(2)^-} = -\sqrt{\frac{1}{2}}\left(\mathbf{e}_k - \mathbf{e}_l\right) & k < l, k, l = 1, 2, \cdots, n \end{cases} \tag{3.82}$$

For the fifth-degree rule, $|\alpha|$ is chosen to be 4 as discussed in *Remark 3.6* and *Theorem 3.4*. Hence,

$$\int_0^\infty r^{n-1+|\alpha|} \exp\left(-r^2\right) dr = \frac{\Gamma\left(n/2+4/2\right)}{2} = \frac{\Gamma\left(n/2\right) n\left(n+2\right)}{8} \tag{3.83}$$

The weights $w_{s,j}$ are given by

$$\begin{aligned} w_{s,j} &= \bar{W}_j \left\|\bar{\boldsymbol{\gamma}}_j\right\|_2^{|\alpha|} \Big/ \int_0^\infty r^{n-1+|\alpha|} \exp\left(-r^2\right) dr \\ &= \bar{W}_j \left\|\bar{\boldsymbol{\gamma}}_j\right\|_2^4 \Big/ \frac{\Gamma\left(n/2\right) n\left(n+2\right)}{8} \\ &= \begin{cases} \left(\frac{4-n}{18}\right) \Big/ \frac{\Gamma\left(n/2\right) n\left(n+2\right)}{8} \pi^{n/2} \cdot \left(\frac{\sqrt{3}}{\sqrt{2}}\right)^4 & i = 1, \cdots, 2n \\ \frac{1}{36} \Big/ \frac{\Gamma\left(n/2\right) n\left(n+2\right)}{8} \pi^{n/2} \left(\frac{\sqrt{6}}{\sqrt{2}}\right)^4 & i = 2n+1, \cdots 2n^2 \end{cases} \\ &= \begin{cases} \pi^{n/2}\left(4-n\right) \Big/ \left(\Gamma\left(n/2\right) n\left(n+2\right)\right) & i = 1, \cdots, 2n \\ 2\pi^{n/2} \Big/ \Gamma\left(n/2\right) n\left(n+2\right) & i = 2n+1, \cdots 2n^2 \end{cases} \\ &= \begin{cases} A_n\left(4-n\right) \Big/ \left(2n\left(n+2\right)\right) & i = 1, \cdots, 2n \\ A_n \Big/ \left(n\left(n+2\right)\right) & i = 2n+1, \cdots 2n^2 \end{cases} \end{aligned} \tag{3.84}$$

Hence,

$$\begin{aligned} &\int_{U_n,5} \boldsymbol{g}\left(\mathbf{s}\right) d\mathbf{s} = \frac{\left(4-n\right) A_n}{2n\left(n+2\right)} \sum_{j=1}^{n} \left(\boldsymbol{g}\left(\mathbf{s}_j^{(1)^+}\right) + \boldsymbol{g}\left(\mathbf{s}_j^{(1)^-}\right)\right) + \\ &\frac{A_n}{n\left(n+2\right)} \sum_{j=1}^{n(n-1)/2} \left(\boldsymbol{g}\left(\mathbf{s}_{j1}^{(2)^+}\right) + \boldsymbol{g}\left(\mathbf{s}_{j1}^{(2)^-}\right) + \boldsymbol{g}\left(\mathbf{s}_{j2}^{(2)^+}\right) + \boldsymbol{g}\left(\mathbf{s}_{j2}^{(2)^-}\right)\right) \end{aligned} \tag{3.85}$$

By combining the fifth-degree spherical rule with the fifth-degree radial rule, the fifth-degree cubature rule can be obtained.

For the fifth-degree radial rule, the following equations must be satisfied.

$$\begin{cases} w_{r,1}r_1^0 + w_{r,2}r_2^0 = \frac{1}{2}\Gamma\left(\frac{1}{2}n\right) \\ w_{r,1}r_1^2 + w_{r,2}r_2^2 = \frac{1}{2}\Gamma\left(\frac{1}{2}n+1\right) = \frac{n}{4}\Gamma\left(\frac{1}{2}n\right) \\ w_{r,1}r_1^4 + w_{r,2}r_2^4 = \frac{1}{2}\Gamma\left(\frac{1}{2}n+2\right) = \frac{1}{2}\left(\frac{1}{2}n+1\right)\left(\frac{1}{2}n\right)\Gamma\left(\frac{1}{2}n\right) \end{cases} \tag{3.86}$$

There are three equations and four variables in Eq. (3.86). Hence, there is one free variable. We choose r_1 as the free variable and set it to 0. Solving these three equations, the points and weights for the fifth-degree radial rule are given in Eq. (3.87) and (3.88), respectively.

$$\begin{cases} r_1 = 0 \\ r_2 = \sqrt{\frac{1}{2}n+1} \end{cases} \tag{3.87}$$

and

$$\begin{cases} w_{r,1} = \frac{1}{2}\Gamma\left(\frac{1}{2}n\right) - \frac{n\Gamma\left(\frac{1}{2}n\right)}{2(n+2)} = \left(\frac{1}{(n+2)}\right)\Gamma\left(\frac{1}{2}n\right) \\ w_{r,2} = \frac{\frac{n}{4}\Gamma\left(\frac{1}{2}n\right)}{\left(\frac{n}{2}+1\right)} = \frac{n}{2(n+2)}\Gamma\left(\frac{1}{2}n\right) \end{cases} \tag{3.88}$$

Note that r_1 may take on other values bedsides 0. However, the number of points for the final cubature rule when r_1 is set to 0 is much less than the number of points when r_1 is set to other values.

Combining Eqs. (2.61), (3.85), (3.87), and (3.88), the 5th-degree cubature rule is given by

$$
\begin{aligned}
&\int_{\mathbb{R}^n} \boldsymbol{g}(\mathbf{x}) N(\mathbf{x};\mathbf{0},\mathbf{I})\, d\mathbf{x} \approx \frac{1}{\pi^{n/2}} \sum_{i=1}^{N_r} \sum_{j=1}^{N_s} w_{r,i} w_{s,j} \boldsymbol{g}\left(\sqrt{2} r_i \mathbf{s}_j\right) \\
&= \frac{2}{n+2} \boldsymbol{g}(\mathbf{0}) + \frac{4-n}{2(n+2)^2} \sum_{j=1}^{n} \left(\boldsymbol{g}\left(\sqrt{n+2} \times \mathbf{s}_j^{(1)^+}\right) + \boldsymbol{g}\left(\sqrt{n+2} \times \mathbf{s}_j^{(1)^-}\right) \right) \\
&+ \frac{1}{(n+2)^2} \sum_{j=1}^{n(n-1)/2} \left(\boldsymbol{g}\left(\sqrt{n+2} \times \mathbf{s}_{j1}^{(2)^+}\right) + \boldsymbol{g}\left(\sqrt{n+2} \times \mathbf{s}_{j1}^{(2)^-}\right) \right) \\
&+ \frac{1}{(n+2)^2} \sum_{j=1}^{n(n-1)/2} \left(\boldsymbol{g}\left(\sqrt{n+2} \times \mathbf{s}_{j2}^{(2)^+}\right) + \boldsymbol{g}\left(\sqrt{n+2} \times \mathbf{s}_{j2}^{(2)^-}\right) \right)
\end{aligned}
\tag{3.89}
$$

3.3 Anisotropic Sparse-grid Quadrature Nonlinear Filter

The anisotropic sparse-grid quadrature (ASGQ) can be used to further improve the computation efficiency of the conventional SGQ. The conventional SGQ is isotropic in the sense that all dimensions are assumed to be equally important and it uses an isotropic sparse-grid, which may result in more points than necessary. There are many practical problems in which different dimensions are not equally important. Motivated by this observation, the ASGQ provides a mechanism for distributing more quadrature points or weights on more important dimensions and allows for better trade-offs between the computational efficiency and the estimation accuracy. The number of ASGQ points can be flexibly controlled by a tunable importance vector and is in general less than that of the SGQ.

3.3.1 Anisotropic Sparse-grid Quadrature

The anisotropic sparse-grid quadrature (ASGQ) is an extension of the conventional sparse-grid method, given by [89]

$$
I_{n,L^{\boldsymbol{\alpha}}}^{\boldsymbol{\alpha}}(\boldsymbol{f}) = \sum_{\Xi^{\boldsymbol{\alpha}} \in \Upsilon_{n,L^{\boldsymbol{\alpha}}}^{\boldsymbol{\alpha}}} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_n}\right)(\boldsymbol{f}) \tag{3.90}
$$

with the accuracy level set

$$\Upsilon^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}} \triangleq \left\{ \Xi^{\boldsymbol{\alpha}} \subset \mathbb{N}^n : i_j \geq 1, \sum_{j=1}^{n} \left(i_j - 1\right)\alpha_j \leq \left(L^{\boldsymbol{\alpha}} - 1\right)\underline{\alpha} \right\}, \ \underline{\alpha} \triangleq \min\left(\boldsymbol{\alpha}\right). \quad (3.91)$$

In Eq. (3.91), $\Xi^{\boldsymbol{\alpha}} \triangleq \left(i_1, \cdots, i_n\right)$ and the superscript „$\boldsymbol{\alpha}$" denotes the anisotropic sparse-grid associated with the importance vector $\boldsymbol{\alpha} = \left[\alpha_1, \cdots, \alpha_j, \cdots, \alpha_n\right] \ \left(\alpha_j > 0\right)$, in which each element of $\boldsymbol{\alpha}$ represents the relative importance of the corresponding state variable and $\min\left(\boldsymbol{\alpha}\right)$ denotes the minimum element in $\boldsymbol{\alpha}$. Without loss of generality, $\underline{\alpha}$ is set to 1. The accuracy level of the ASGQ is denoted by $L^{\boldsymbol{\alpha}} \in \mathbb{N}$.

Equation (3.90) can be rewritten as [89]

$$I^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}}\left(\boldsymbol{f}\right) = \sum_{\Xi^{\boldsymbol{\alpha}} \in \mathbf{Y}^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}}} c^{\boldsymbol{\alpha}}\left(\Xi^{\boldsymbol{\alpha}}\right)\left(I_{i_1} \otimes \cdots \otimes I_{i_n}\right)\left(\boldsymbol{f}\right) \quad (3.92)$$

where

$$c^{\boldsymbol{\alpha}}\left(\Xi^{\boldsymbol{\alpha}}\right) \triangleq \sum_{\Psi \in \{0,1\}^n, \Xi^{\boldsymbol{\alpha}} + \Psi \in \Upsilon^{\boldsymbol{\alpha}}_{n,L^{\alpha}}} \left(-1\right)^{|\Psi|} \quad (3.93)$$

$$\mathbf{Y}^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}} \triangleq \Upsilon^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}} \setminus \Upsilon^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}} - \frac{|\boldsymbol{\alpha}|}{\underline{\alpha}}} \quad (3.94)$$

Note that $\{0,1\}^n$ denotes the set of all *n*-dimensional sequences with each dimension"s value being 0 or 1. $|\Psi|$ and $|\boldsymbol{\alpha}|$ denote the summation of the elements in Ψ and $\boldsymbol{\alpha}$ respectively; „\" denotes the subtraction operation of two sets. Equations (3.92)-(3.93) are valid whether $\boldsymbol{\alpha}$ is an integer vector or not, but for the convenience of formulation, $\boldsymbol{\alpha}$ is assumed to be an integer vector such that $\frac{|\boldsymbol{\alpha}|}{\underline{\alpha}}$ and $L^{\boldsymbol{\alpha}} - \frac{|\boldsymbol{\alpha}|}{\underline{\alpha}}$ are integers.

By Eqs. (3.91) and (3.94),

$$\mathbf{Y}^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}} = \Upsilon^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}} \setminus \Upsilon^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}} - \frac{|\boldsymbol{\alpha}|}{\underline{\alpha}}} = \left\{ \Xi^{\boldsymbol{\alpha}} \subset \mathbb{N}^n ; i_j \geq 1, \left(L^{\boldsymbol{\alpha}} - \frac{|\boldsymbol{\alpha}|}{\underline{\alpha}} - 1 \right)\underline{\alpha} < \sum_{j=1}^{n} \left(i_j - 1\right)\alpha_j \leq \left(L^{\boldsymbol{\alpha}} - 1\right)\underline{\alpha} \right\} \quad (3.95)$$

From Eq. (3.92), the corresponding ASGQ point set defined by $\mathbf{X}^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}}$ is

$$\mathbf{X}^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}} = \bigcup_{\Xi^{\boldsymbol{\alpha}} \in \mathbf{Y}^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}}} \left(X_{i_1} \otimes \cdots \otimes X_{i_n}\right) \quad (3.96)$$

The weight W_i for each point $\boldsymbol{\gamma}_i$ in $\mathbf{X}^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}}$ is the sum of the weights on the point over all combinations of $X_{i_1} \otimes \cdots \otimes X_{i_n}$ containing the point. Moreover, for one specific combination, the weight on the point $\boldsymbol{\gamma}_i$ is calculated by $W_i = c^{\boldsymbol{\alpha}}\left(\Xi^{\boldsymbol{\alpha}}\right)\left(w_{s_1} \times \cdots \times w_{s_j} \times \cdots \times w_{s_n}\right)$.

Remark 3.7: Given the accuracy level- $L^{\boldsymbol{\alpha}}$ and dimension *n*, the final ASGQ point set is determined by $\mathbf{Y}^{\boldsymbol{\alpha}}_{n,L^{\boldsymbol{\alpha}}}$. When the importance vector $\boldsymbol{\alpha}$ is also chosen, the upper bound of the inequality in Eq. (3.95) is fixed. This implies that for a larger α_j, the allowable range of i_j is smaller. In other words, if $\alpha_i < \alpha_j$, by Eq. (3.95), it can be shown that $\max(i_i) > \max(i_j)$, which indicates that the ASGQ can use a higher accuracy level univariate quadrature rule $I_{\max(i_i)}$ for dimension *i* than the univariate quadrature rule $I_{\max(i_j)}$ for dimension *j*. Note that the higher accuracy level univariate quadrature rule uses more points than the lower accuracy level quadrature rule. Hence, a smaller element in $\boldsymbol{\alpha}$ indicates that the corresponding dimension is more important and uses more quadrature points.

Remark 3.8: By Eqs. (3.95) and (3.96), the effect of $\boldsymbol{\alpha}$ on the ASGHQ point set is determined by the ratio of $\alpha_j/\underline{\alpha}$ or $\underline{\alpha}/\alpha_j$ rather than the absolute value of α_j.

To better illustrate the ASGQ rule and compare it with the SGQ, $n = 2$, $L^{\boldsymbol{\alpha}} = 3$ are used as an example to show how to construct the ASGQ point set. Assume $\boldsymbol{\alpha} = [2,1]$. Then, $\boldsymbol{\Upsilon}^{\boldsymbol{\alpha}}_{2,3} = \left\{\Xi^{\boldsymbol{\alpha}} \subset \mathbb{N}^2 : i_j \geq 1, 2(i_1 - 1) + (i_2 - 1) \leq 2\right\} = \left\{\Xi^{\boldsymbol{\alpha}} \subset \mathbb{N}^2 : i_j \geq 1, i_1 + 0.5 i_2 \leq 2.5\right\}$, It can be rewritten as $\boldsymbol{\Upsilon}^{\boldsymbol{\alpha}}_{2,3} = \{(1,1),(1,2),(2,1),(1,3)\}$.

By Eq. (3.95), $\mathbf{Y}^{\boldsymbol{\alpha}}_{2,3} = \boldsymbol{\Upsilon}^{\boldsymbol{\alpha}}_{2,3} \setminus \boldsymbol{\Upsilon}^{\boldsymbol{\alpha}}_{2,3-\frac{|\boldsymbol{\alpha}|}{\underline{\alpha}}}$

and $\boldsymbol{\Upsilon}^{\boldsymbol{\alpha}}_{2,3-\frac{|\boldsymbol{\alpha}|}{\underline{\alpha}}} = \boldsymbol{\Upsilon}^{\boldsymbol{\alpha}}_{2,0} = \left\{\Xi^{\boldsymbol{\alpha}} \subset \mathbb{N}^2; i_j \geq 1, i_1 + 0.5 i_2 \leq 1\right\} = \varnothing$.

Thus, $\mathbf{Y}^{\boldsymbol{\alpha}}_{2,3} = \left\{\Xi^{\boldsymbol{\alpha}} \subset \mathbb{N}^2 : i_j \geq 1, 1 < i_1 + 0.5 i_2 \leq 2.5\right\} = \{(1,1),(1,2),(2,1),(1,3)\}$.

By Eq. (3.93), $c^{\boldsymbol{\alpha}}\left((1,1)\right)=-1$, $c^{\boldsymbol{\alpha}}\left((1,2)\right)=0$, $c^{\boldsymbol{\alpha}}\left((2,1)\right)=1$ and $c^{\boldsymbol{\alpha}}\left((1,3)\right)=1$. Then, $I_{2,3}^{\boldsymbol{\alpha}}=-\left(I_1\otimes I_1\right)+\left(I_2\otimes I_1\right)+\left(I_1\otimes I_3\right)$.

The final ASGQ point set is shown in Figure 3.4. Note that Figure 3.4 verifies the conclusion of *Remark 3.7*. That is, the value „2" in $\boldsymbol{\alpha}=[2,1]$ makes i_1 take a smaller range of values than i_2, and thus generates fewer points in the vertical dimension than the value „1" in $\boldsymbol{\alpha}$ generates in the horizontal dimension.

Compared with the conventional SGQ, ASGQ does not contain the tensor product $X_1\otimes X_2$, owing to the fact that $c^{\boldsymbol{\alpha}}\left((1,2)\right)=0$, and also does not contain $X_2\otimes X_2$ and $X_3\otimes X_1$ owing to the fact that $\{(2,2),(3,1)\}\not\subset \mathbf{Y}_{2,3}^{\boldsymbol{\alpha}}=\{(1,1),(1,2),(2,1),(1,3)\}$.

Note that the actual accuracy of the level-L^{α} ASGQ may be different from that of the level-L SGQ when $L=L^{\alpha}$. For example, when $L^{\alpha}=L=3$ and $n=2$, by *Theorem 3.1*, SGQ can calculate the polynomial $x_1^2x_2^2$ exactly, whereas the ASGQ cannot, as seen from the above expression of $I_{2,3}^{\boldsymbol{\alpha}}$.

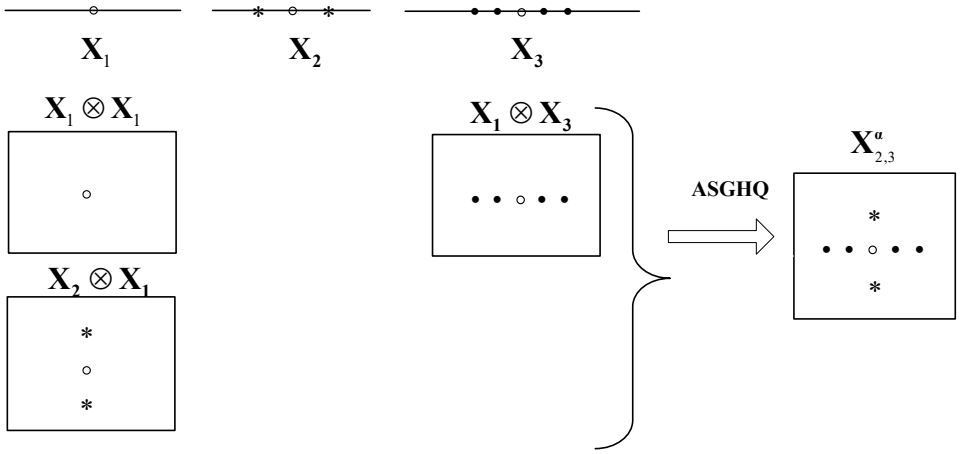

Figure 3.4 ASGQ with the accuracy level-3 and $\boldsymbol{\alpha}=[2,1]$ for 2-dimensional problems

3.3.2 Analysis of the Accuracy of the Anisotropic Sparse-grid Quadrature

The conventional SGQ can be viewed as a special case of the ASGQ. When all elements in $\boldsymbol{\alpha}$ are the same, Eq. (3.92) and Eq. (3.96) reduce to Eq. (3.7) and Eq. (3.12), respectively, and the ASGQ reduces to the SGQ [89]. Note that in Eq. (3.95) and Eq. (3.8), when all elements in $\boldsymbol{\alpha}$ are the same, $L-n-1<\sum_{j=1}^{n}\left(i_j-1\right)$ is the same as $L-n\le\sum_{j=1}^{n}\left(i_j-1\right)$.

Remark 3.9: If $\left(\frac{\alpha_j}{\underline{\alpha}}\right)>L^{\boldsymbol{\alpha}}-1$, by Eq. (3.95), i_j must be 1, which implies that there will be only one point in dimension j, a case of degenerated accuracy. To prevent this case from happening, a constraint should be satisfied for the ratio, i. e. $1\le\left(\frac{\alpha_j}{\underline{\alpha}}\right)\le L^{\boldsymbol{\alpha}}-1$. Under this constraint, the level-2 ($L^{\boldsymbol{\alpha}}=2$) ASGQ is the same as the level-2 ($L=2$) SGQ since $\frac{\alpha_j}{\underline{\alpha}}=1$, and every α_j is the same. Note that if $L^{\boldsymbol{\alpha}}=L$ and all elements in $\boldsymbol{\alpha}$ are the same, by Eq. (3.95), the maximum value of i_j is $L^{\boldsymbol{\alpha}}$ in each dimension, which becomes SGQ. In most cases, $\alpha_j/\underline{\alpha}>1$, by Eq. (3.95), $i_j<L^{\boldsymbol{\alpha}}$, which means ASGQ is less accurate than SGQ in dimension j in these cases.

The relationship between the ASGQ and the SGQ can be revealed by the following proposition and theorem.

Proposition 3.3: If $L^{\alpha}=L$ and $L-n\le 0$, the ASGQ point set is a subset of the SGQ point set.

Proof: By Eq. (3.95), $\mathbf{Y}^{\boldsymbol{\alpha}}_{n,L^{\alpha}}$ can be rewritten as

$$\mathbf{Y}^{\boldsymbol{\alpha}}_{n,L^{\alpha}}=\left\{\Xi^{\boldsymbol{\alpha}}\subset\mathbb{N}^n;i_j\ge 1,\left(L^{\boldsymbol{\alpha}}-1\right)-\frac{|\boldsymbol{\alpha}|}{\underline{\alpha}}<\sum_{j=1}^{n}\left(i_j-1\right)\frac{\alpha_j}{\underline{\alpha}}\le\left(L^{\boldsymbol{\alpha}}-1\right)\right\}.$$

If $L-n\le 0$, $\left(L^{\alpha}-1\right)-\frac{|\boldsymbol{\alpha}|}{\underline{\alpha}}\le\left(L^{\alpha}-1\right)-n=L^{\alpha}-n-1<0$. Since $\sum_{j=1}^{n}\left(i_j-1\right)\frac{\alpha_j}{\underline{\alpha}}\ge 0$,

$\mathbf{Y}^{\boldsymbol{\alpha}}_{n,L^{\alpha}}=\left\{\Xi^{\boldsymbol{\alpha}}\subset\mathbb{N}^n;i_j\ge 1,0\le\sum_{j=1}^{n}\left(i_j-1\right)\frac{\alpha_j}{\underline{\alpha}}\le\left(L^{\boldsymbol{\alpha}}-1\right)\right\}$.

Recall that $\mathbf{Y}_{n,L}=\left\{\Xi\subset\mathbb{N}^n;i_j\ge 1,L-n\le\sum_{j=1}^{n}\left(i_j-1\right)\le\left(L-1\right)\right\}$ for the SGQ. Since $L-n\le 0$ and $\sum_{j=1}^{n}\left(i_j-1\right)\ge 0$, $\mathbf{Y}_{n,L}=\left\{\Xi\subset\mathbb{N}^n;i_j\ge 1,0\le\sum_{j=1}^{n}\left(i_j-1\right)\le\left(L-1\right)\right\}$. Comparing the above $\mathbf{Y}^{\boldsymbol{\alpha}}_{n,L^{\alpha}}$ with $\mathbf{Y}_{n,L}$ shows $\mathbf{Y}^{\boldsymbol{\alpha}}_{n,L^{\alpha}}\subset\mathbf{Y}_{n,L}$ and thus $\mathbf{X}^{\boldsymbol{\alpha}}_{n,L^{\alpha}}\subset\mathbf{X}_{n,L}$ because $\frac{\alpha_j}{\underline{\alpha}}\ge 1$.■

Remark 3.10: The condition $L-n\le 0$ is needed for *Proposition* 3.3. For example, if $L^{\alpha}=L=6$, $n=2$ and $\boldsymbol{\alpha}=[1,2]$, which violates the condition, it can be verified that some points in ASGQ are not included in the SGQ point set.

It is worth noting that for typical multidimensional problems, the accuracy level-*L* is usually less than the dimension *n*. So the condition in *Proposition* 3.3 is satisfied in most cases.

The ASGQ accuracy depends on the accuracy level-$L^{\boldsymbol{\alpha}}$ and the parameter $\boldsymbol{\alpha}$, whereas the SGQ accuracy only depends on the accuracy level-*L*. The accuracy of the ASGQ is guaranteed by the following theorem.

Theorem 3.6: If $\boldsymbol{\alpha}=\left\lfloor 1,\cdots,1,\underbrace{\alpha_m,\cdots,\alpha_m}_{n_1},\underbrace{\alpha_m,\cdots,\alpha_m}_{n_2}\right\rfloor,(n_1+n_2=n)$ and $2\le\alpha_m+1\le L^{\boldsymbol{\alpha}}$, ASGQ is either more accurate than or as accurate as the level-2 SGQ.

Proof: For this $\boldsymbol{\alpha}$,

$\Upsilon^{\boldsymbol{\alpha}}_{n_1+n_2,L^{\boldsymbol{\alpha}}}=\left\{\Xi^{\boldsymbol{\alpha}}\subset\mathbb{N}^{n_1+n_2};i_j\ge 1,\sum_{j=1}^{n_1}\left(i_j-1\right)+\alpha_m\sum_{j=n_1+1}^{n_1+n_2}\left(i_j-1\right)\le\left(L^{\boldsymbol{\alpha}}-1\right)\right\}$. Since $i_j\ge 1$,

$0\le\sum_{j=n_1+1}^{n_1+n_2}\left(i_j-1\right)\le\frac{1}{\alpha_m}\left(\left(L^{\boldsymbol{\alpha}}-1\right)-\sum_{j=1}^{n_1}\left(i_j-1\right)\right)$. Hence, each $i_j\left(j=n_1+1,\cdots,n_1+n_2\right)$ can be

$1,\cdots,\left\lfloor \frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}+1 \right\rfloor$, where „$\lfloor\ \rfloor$“ denotes the floor operation, which returns an integer less than or equal to $\frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}+1$. Note that $\alpha_m+1\le L^{\boldsymbol{\alpha}}$ or $\alpha_m \le L^{\boldsymbol{\alpha}}-1$ implies $\frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}+1\ge 2$.

There are three possible cases with this $\boldsymbol{\alpha}$:

Case (1): All $i_j\left(j=1,\cdots,n_1\right)$ are 1 and $\sum_{j=n_1+1}^{n_1+n_2}\left(i_j-1\right)\le\frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}$. Define this set of $\Xi^{\boldsymbol{\alpha}}$ as

$$\Upsilon_{s1}=\left\{\Xi^{\boldsymbol{\alpha}}\subset\Upsilon^{\boldsymbol{\alpha}}_{n_1+n_2,L^{\boldsymbol{\alpha}}};i_j\ge 1,\ \sum_{j=1}^{n_1}\left(i_j-1\right)=0,\quad 0\le\sum_{j=n_1+1}^{n_1+n_2}\left(i_j-1\right)\le\frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}\right\} \tag{3.97}$$

By Eq. (3.6), the range of $i_j\left(j=n_1+1,\cdots,n_1+n_2\right)$ in Υ_{s1} is identical to that of

$$\Upsilon_{n_2,\left\lfloor\frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}+1\right\rfloor}=\left\{\Xi\subset\mathbb{N}^{n_2};i_j\ge 1,\sum_{j=n_1+1}^{n_1+n_2}\left(i_j-1\right)\le\left\lfloor\frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}\right\rfloor\right\} \tag{3.98}$$

for the SGQ.

Case (2): All $i_j\left(j=n_1+1,\cdots,n_1+n_2\right)$ are 1 and $\sum_{j=1}^{n_1}\left(i_j-1\right)\le L^{\boldsymbol{\alpha}}-1$. Define this set of $\Xi^{\boldsymbol{\alpha}}$ as

$$\Upsilon_{s2}=\left\{\Xi^{\boldsymbol{\alpha}}\subset\Upsilon^{\boldsymbol{\alpha}}_{n_1+n_2,L^{\boldsymbol{\alpha}}};i_j\ge 1,\quad 0\le\sum_{j=1}^{n_1}\left(i_j-1\right)\le L^{\boldsymbol{\alpha}}-1,\quad\sum_{j=n_1+1}^{n_1+n_2}\left(i_j-1\right)=0\right\}. \tag{3.99}$$

Similarly, by Eq. (3.6), the range of $i_j\left(j=1,\cdots,n_1\right)$ in Υ_{s2} is identical to that of

$$\Upsilon_{n_1,L^{\boldsymbol{\alpha}}}=\left\{\Xi\subset\mathbb{N}^{n_1};i_j\ge 1,\sum_{j=1}^{n_1}\left(i_j-1\right)\le L^{\boldsymbol{\alpha}}-1\right\} \tag{3.100}$$

for the SGQ.

Note that Case (1) and Case (2) contain a duplicate case when all $i_j=1,\left(j=1,\cdots,n_1+n_2\right)$, i.e.

$$\Upsilon_{s3}\triangleq\Upsilon_{s1}\bigcap\Upsilon_{s2}=\left\{\Xi^{\boldsymbol{\alpha}}\subset\Upsilon^{\boldsymbol{\alpha}}_{n_1+n_2,L^{\boldsymbol{\alpha}}};i_j\ge 1,\sum_{j=1}^{n_1+n_2}\left(i_j-1\right)=0\right\} \tag{3.101}$$

Case (3): $\sum_{j=1}^{n_1}\left(i_j-1\right)\ge 1$ and $1\le\sum_{j=n_1+1}^{n_1+n_2}\left(i_j-1\right)\le\left\lfloor\frac{L^{\boldsymbol{\alpha}}-2}{\alpha_m}\right\rfloor$. Define this set of $\Xi^{\boldsymbol{\alpha}}$ as

$$\Upsilon_{s4}=\left\{\Xi^{\boldsymbol{\alpha}}\subset\Upsilon^{\boldsymbol{\alpha}}_{n_1+n_2,L^{\boldsymbol{\alpha}}};i_j\geq 1,\ 1\leq\sum_{j=1}^{n_1}\left(i_j-1\right)\leq L^{\boldsymbol{\alpha}}-\alpha_m-1,\quad 1\leq\sum_{j=n_1+1}^{n_1+n_2}\left(i_j-1\right)\leq\left\lfloor\frac{L^{\boldsymbol{\alpha}}-2}{\alpha_m}\right\rfloor\right\} \quad (3.102)$$

Note that $\Upsilon_{s1}\bigcap\Upsilon_{s4}=\varnothing$ and $\Upsilon_{s2}\bigcap\Upsilon_{s4}=\varnothing$.

In summary, $\Upsilon^{\boldsymbol{\alpha}}_{n_1+n_2,L^{\boldsymbol{\alpha}}}=\Upsilon_{s1}\bigcup\Upsilon_{s2}\setminus\Upsilon_{s3}\bigcup\Upsilon_{s4}$. Hence,

$$\begin{aligned}
I^{\boldsymbol{\alpha}}_{n_1+n_2,L^{\boldsymbol{\alpha}}} &= \sum_{\Xi^{\boldsymbol{\alpha}}\in\Upsilon^{\boldsymbol{\alpha}}_{n_1+n_2,L^{\boldsymbol{\alpha}}}}\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\otimes\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)\\
&= \sum_{\Xi^{\boldsymbol{\alpha}}\in(\Upsilon_{s1}\cup\Upsilon_{s2}\setminus\Upsilon_{s3}\cup\Upsilon_{s4})}\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\otimes\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)\\
&= \left(\underbrace{\Delta^1\otimes\cdots\otimes\Delta^1}_{n_1\text{ elements}}\otimes\sum_{\Xi\in\Upsilon_{n_2,\left\lfloor\frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}+1\right\rfloor}}\left(\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)\right)+\left(\sum_{\Xi\in\Upsilon_{n_1,L^{\boldsymbol{\alpha}}}}\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\right)\otimes\underbrace{\Delta^1\cdots\otimes\Delta^1}_{n_2\text{ elements}}\right)\\
&\quad-\underbrace{\Delta^1\otimes\Delta^1\cdots\otimes\Delta^1}_{n_1+n_2}+\sum_{\Xi^{\boldsymbol{\alpha}}\in\Upsilon_{s4}}\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\otimes\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)\\
&= \left(\underbrace{I_1\otimes\cdots\otimes I_1}_{n_1\text{ elements}}\otimes\sum_{\Xi\in\Upsilon_{n_2,\left\lfloor\frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}+1\right\rfloor}}\left(\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)\right)+\left(\sum_{\Xi\in\Upsilon_{n_1,L^{\boldsymbol{\alpha}}}}\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\right)\otimes\underbrace{I_1\cdots\otimes I_1}_{n_2\text{ elements}}\right)\\
&\quad-\underbrace{I_1\otimes I_1\cdots\otimes I_1}_{n_1+n_2}+\sum_{\Xi^{\boldsymbol{\alpha}}\in\Upsilon_{s4}}\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\otimes\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)\\
&= I_{n_1,1}\otimes I_{n_2,\left\lfloor\frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}+1\right\rfloor}+I_{n_1,L^{\boldsymbol{\alpha}}}\otimes I_{n_2,1}-I_{n_1+n_2,1}+\sum_{\Xi^{\boldsymbol{\alpha}}\in\Upsilon_{s4}}\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\otimes\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)
\end{aligned} \quad (3.103)$$

From the 3rd equality to the 4th equality in Eq. (3.103), the fact that $\Delta^1=I_1-I_0=I_1$ is used. Note that $\underbrace{I_1\otimes\cdots\otimes I_1}_{n_1\text{ elements}}\otimes\sum_{\Xi\in\Upsilon_{n_2,\left\lfloor\frac{L^{\boldsymbol{\alpha}}-1}{\alpha_m}+1\right\rfloor}}\left(\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)$ and $\sum_{\Xi\in\Upsilon_{n_1,L^{\boldsymbol{\alpha}}}}\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\right)\otimes\underbrace{I_1\cdots\otimes I_1}_{n_2\text{ elements}}$ correspond to Case (1) and Case (2) respectively. $\underbrace{I_1\otimes I_1\cdots\otimes I_1}_{n_1+n_2}$ is the duplicate part of Case (1) and Case (2). The last term $\sum_{\Xi^{\boldsymbol{\alpha}}\in\Upsilon_{s4}}\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\otimes\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)$ corresponds to Case (3) and has no duplicate

part with Case (1) and Case (2). It is worth noting that when $L^{\alpha} < 2+\alpha_m$, this last term vanishes and Υ_{s4} is empty, as seen from Eq. (3.102).

Now, compare Eq. (3.103) with the level-2 SGQ. By Eqs. (3.4) and (3.6), the level-2 SGQ is $I_{n_1+n_2,2} = \sum_{\Xi\in\Upsilon_{n_1+n_2,2}} \left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\otimes\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)$ and $\sum_{j=1}^{n_1+n_2}\left(i_j-1\right)\le 1$. If all $i_j\left(j=1,\cdots,n_1\right)$ are 1, then $\sum_{j=n_1+1}^{n_1+n_2}\left(i_j-1\right)\le 1$; if $\sum_{j=1}^{n_1}\left(i_j-1\right)\le 1$, all $i_j\left(j=n_1+1,\cdots,n_1+n_2\right)$ must be 1. Note that both cases contain the duplicate case when all $i_j=1\left(j=1,\cdots,n_1+n_2\right)$. As with Eq. (3.103), these two cases lead to

$$
\begin{aligned}
I_{n_1+n_2,2} &= \sum_{\Xi\in\Upsilon_{n,2}} \left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\otimes\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right) \\
&= \left(\underbrace{I_1\otimes\cdots\otimes I_1}_{n_1 \text{ elements}}\otimes\sum_{\Xi\in\Upsilon_{n_2,2}}\left(\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)\right)+\left(\sum_{\Xi\in\Upsilon_{n_1,2}}\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_{n_1}}\right)\otimes\underbrace{I_1\cdots\otimes I_1}_{n_2 \text{ elements}}\right) \\
&\quad -\underbrace{I_1\otimes I_1\cdots\otimes I_1}_{n_1+n_2} \\
&= I_{n_1,1}\otimes I_{n_2,2}+I_{n_1,2}\otimes I_{n_2,1}-I_{n_1+n_2,1}
\end{aligned}
\tag{3.104}
$$

Remark 3.11: By *Theorem 3.1*, $I_{n_1+n_2,2}$ is exact for all polynomials $x_1^{a_1}\cdots x_j^{a_j}\cdots x_{n_1+n_2}^{a_{n_1+n_2}}$ with $0\le\sum_{j=1}^{n_1+n_2}a_j\le 3$. Note, however, that the condition of *Theorem 3.6* is sufficient but not necessary. In fact, from Eq. (3.104), it can be seen that $I_{n_1+n_2,2}$ is exact for all polynomials $x_1^{a_1}\cdots x_j^{a_j}\cdots x_{n_1+n_2}^{a_{n_1+n_2}}$ with $0\le\sum_{j=1}^{n_1}a_j\le 1, 0\le\sum_{j=n_1+1}^{n_1+n_2}a_j\le 3$ because of the exactness of $I_{n_1,1}\otimes I_{n_2,2}$ for these polynomials. $I_{n_1+n_2,2}$ is also exact for all polynomials $x_1^{a_1}\cdots x_j^{a_j}\cdots x_{n_1+n_2}^{a_{n_1+n_2}}$ with $0\le\sum_{j=1}^{n_1}a_j\le 3, 0\le\sum_{j=n_1+1}^{n_1+n_2}a_j\le 1$, because of the exactness of $I_{n_1,2}\otimes I_{n_2,1}$ is exact for these polynomials. The above discussions also apply to the UT, which is exact up to 3rd order polynomials [67], but may be exact for some higher order polynomials. On the other hand, the exactness of $I_{n_1+n_2,2}$ for the above polynomials

cannot be further generalized to all the polynomials with $0 \le \sum_{j=1}^{n_1+n_2} a_j \le 4$. For example, $I_{n_1+n_2,2}$ is not exact for all polynomials $x_1^{a_1} \cdots x_j^{a_j} \cdots x_{n_1+n_2}^{a_{n_1+n_2}}$ with $0 \le \sum_{j=1}^{n_1} a_j \le 2, 0 \le \sum_{j=n_1+1}^{n_1+n_2} a_j \le 2$. Note that the UT has the same accuracy as the level-2 SGQ [25] because all polynomials for which the UT is exact can be exactly calculated by the level-2 SGQ.

Comparing Eq. (3.103) with Eq. (3.104), one can see that when $L^{\boldsymbol{\alpha}} > 2$ and $\left\lfloor \frac{L^{\boldsymbol{\alpha}} - 1}{\alpha_m} + 1 \right\rfloor \ge 2$, there exist higher-order polynomials that can be exactly calculated by the first three terms in Eq. (3.103), but cannot by $I_{n_1+n_2,2}$ (level-2 SGQ). When $L^{\boldsymbol{\alpha}} = 2$, $\alpha_m = 1$, the first three terms in Eq. (3.103) are as accurate as $I_{n_1+n_2,2}$ (level-2 SGQ) is.

Now, let us consider the contribution of $\sum_{\Xi^{\boldsymbol{\alpha}} \in \boldsymbol{\Upsilon}_{s4}} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_{n_1}} \otimes \Delta^{i_{n_1+1}} \otimes \cdots \otimes \Delta^{i_{n_1+n_2}} \right)$, the last term of Eq. (3.103), to $I^{\boldsymbol{\alpha}}_{n_1+n_2,L^{\boldsymbol{\alpha}}}$. This term is generated from Case (3) where $1 \le \sum_{j=1}^{n_1} (i_j - 1) \le L^{\boldsymbol{\alpha}} - \alpha_m - 1$, and $1 \le \sum_{j=n_1+1}^{n_1+n_2} (i_j - 1) \le \left\lfloor \frac{L^{\boldsymbol{\alpha}} - 2}{\alpha_m} \right\rfloor$. When both inequalities are satisfied, there exist at least one $i_m \ge 2 \, (1 \le m \le n_1)$ and one $i_k \ge 2 \, (n_1 + 1 \le k \le n_1 + n_2)$.

For the polynomials that can be exactly calculated by $I_{n_1,s} \otimes I_{n_2,1}$ $(s \in \mathbb{N})$ using Eq. (3.70), the contribution of the last term in Eq. (3.103) is 0, i.e.

$$\begin{aligned} &\left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_m} \otimes \cdots \otimes \Delta^{i_{n_1}} \right) \otimes \left(\Delta^{i_{n_1+1}} \otimes \cdots \otimes \Delta^{i_k} \otimes \cdots \otimes \Delta^{i_{n_1+n_2}} \right) \\ &= \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_m} \otimes \cdots \otimes \Delta^{i_{n_1}} \right) \otimes \left(\Delta^{i_{n_1+1}} \otimes \cdots \otimes 0 \otimes \cdots \otimes \Delta^{i_{n_1+n_2}} \right) \\ &= 0 \end{aligned} \tag{3.105}$$

That is because a polynomial f that can be exactly calculated by I_1 can be exactly calculated by $I_j \, (j \ge 1)$ as well, i.e. $I_j(f) = I_1(f) \, (j \ge 1)$ and thus $\Delta^{i_k}(f) = I_{i_k}(f) - I_{i_k - 1}(f) = 0 \, (i_k \ge 2)$.

Likewise, for the polynomials that can be exactly calculated by $I_{n_1,1} \otimes I_{n_2,s}$ $(s \in \mathbb{N})$ using Eq. (3.103), the contribution of the last term in Eq. (3.103) is also 0, i.e.

$$\begin{aligned}
&\left(\Delta^{i_1}\otimes\cdots\otimes\Delta^{i_m}\otimes\cdots\otimes\Delta^{i_{n_1}}\right)\otimes\left(\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_k}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)\\
&=\left(\Delta^{i_1}\otimes\cdots\otimes 0\otimes\cdots\otimes\Delta^{i_{n_1}}\right)\otimes\left(\Delta^{i_{n_1+1}}\otimes\cdots\otimes\Delta^{i_k}\otimes\cdots\otimes\Delta^{i_{n_1+n_2}}\right)\\
&=0
\end{aligned}\tag{3.106}$$

Hence, any polynomial that can be exactly calculated by $I_{n_1,2}\otimes I_{n_2,1}$, $I_{n_1,1}\otimes I_{n_2,2}$, or $I_{n_1+n_2,2}$ can be exactly calculated by Eq. (3.103). As discussed in *Remark 3.11*, the types of polynomials that can be exactly calculated by $I_{n_1+n_2,2}$ are all covered by the types of polynomials that can be exactly calculated by $I_{n_1,2}\otimes I_{n_2,1}$ or $I_{n_1,1}\otimes I_{n_2,2}$.

To summarize, when $2\le\alpha_m+1\le L^{\boldsymbol{\alpha}}$, because 1) the first three terms in Eq. (3.103) are more accurate than or as accurate as the level-2 SGQ ($I_{n_1+n_2,2}$) and 2) the last term in Eq. (3.103) is 0 for polynomials that can be exactly calculated by $I_{n_1,s}\otimes I_{n_2,1}$ or $I_{n_1,1}\otimes I_{n_2,s}$ ($s\in\mathbb{N}$), which include the first three terms in Eq. (3.103), we may conclude that ASGQ is more accurate than or as accurate as the level-2 SGQ.■

Note that for sufficiently large $L^{\boldsymbol{\alpha}}$, there exist polynomials that can be exactly calculated by the last term in Eq. (3.103) but cannot be exactly calculated by either the first three terms of Eq. (3.103) or the level-2 SGQ ($I_{n_1+n_2,2}$). Examples of these polynomials include $x_m^2x_k^2$ ($1\le m\le n_1,\ n_1+1\le k\le n_1+n_2$). Also recall that when $L^{\boldsymbol{\alpha}}=2$ and $\alpha_m=1$, the ASGQ is identical to the level-2 SGQ as discussed in *Remark 3.9*.

Remark 3.12: As discussed in *Remark 3.11*, the UT has the same accuracy as the level-2 SGQ. Thus, the ASGQ is more accurate than the UT when $2<\alpha_m+1\le L^{\boldsymbol{\alpha}}$.

When $L^{\boldsymbol{\alpha}}=3$ and $\boldsymbol{\alpha}=\left\lfloor 1,\cdots,1,\underbrace{2,\cdots,2}_{n_2}\right\rfloor$, which may be used for many applications, the following equation can be obtained from Eq. (3.103).

$$I^{\boldsymbol{\alpha}}_{n_1+n_2,3}=\left(I_{n_1,1}\otimes I_{n_2,2}\right)+\left(I_{n_1,3}\otimes I_{n_2,1}\right)-I_{n_1+n_2,1}\tag{3.107}$$

The illustration of ASGQ in Figure 3.4 can be viewed as a special case ($n_1=1,n_2=1,L^{\boldsymbol{\alpha}}=3$) of Eq. (3.107).

Note that, by Eq. (3.107), $I^{\boldsymbol{\alpha}}_{n_1+n_2,3}$ is exact for all the polynomials of the form $x_1^{a_1}\cdots x_j^{a_j}\cdots x_{n_1+n_2}^{a_{n_1+n_2}}$ with $0\le\sum_{j=1}^{n_1}a_j\le 1, 0\le\sum_{j=n_1+1}^{n_1+n_2}a_j\le 3$ or $0\le\sum_{j=1}^{n_1}a_j\le 5, 0\le\sum_{j=n_1+1}^{n_1+n_2}a_j\le 1$. In contrast, the UT is exact for all polynomials with $0\le\sum_{j=1}^{n_1+n_2}a_j\le 3$. Therefore, ASGQ is more accurate than UT when $\boldsymbol{\alpha}=\left\lfloor 1,\cdots,1,\underbrace{2,\cdots,2}_{n_2}\right\rfloor$ and $L^{\boldsymbol{\alpha}}=3$, which satisfies the condition of *Theorem 3.6*.

3.3.3 The Anisotropic Sparse-grid Quadrature Filter

The anisotropic sparse-grid quadrature filter algorithm (ASGQF) is designed to place more sparse-grid quadrature points in directions with larger uncertainties according to the state covariance information. In this dissertation, the eigenvalues of the covariance matrix are used as the metric of uncertainty. However, the state variables may have very different scales or units, which make it difficult for the eigenvalues to reflect the true relative uncertainties. To eliminate this effect, the covariance matrix $\mathbf{P}$ is scaled by a diagonal matrix $\mathbf{D}$. The elements of the importance vector $\boldsymbol{\alpha}$, which put weights on a set of mutually perpendicular directions given by the eigenvectors of the scaled covariance matrix $\mathbf{DPD}^T$, are determined based on the eigenvalues of $\mathbf{DPD}^T$. An element of $\boldsymbol{\alpha}$ is assigned a small value or high importance if the corresponding eigenvalue is large.

The anisotropic sparse-grid quadrature filter algorithm (ASGQF) differs from the SGQF algorithm of [25] only in the points $\boldsymbol{\gamma}_i$, the weights W_i, and the covariance decomposition matrix $\mathbf{S}$ in Eq. (2.31) and $\tilde{\mathbf{S}}$ in Eq. (2.35). They are obtained by the following procedure in the ASGQF, where $\mathbf{P}=\mathbf{P}_{k-1|k-1}$ is used for the prediction step and $\mathbf{P}=\mathbf{P}_{k|k-1}$ is used for the update step.

1) Calculate the eigenvalues of $\mathbf{DPD}^T$ using the singular value decomposition (SVD) or eigenvalue decomposition (EIG). In MATLAB, SVD sorts the eigenvalues in descending order and EIG sorts the eigenvalues in ascending order.
2) Determine the importance vector $\boldsymbol{\alpha}$, based on the eigenvalues of $\mathbf{DPD}^T$. The elements of $\boldsymbol{\alpha}$ are in ascending order if the eigenvalues of $\mathbf{DPD}^T$ are in descending order.
3) Generate the ASGQ points $\boldsymbol{\gamma}_i$ and weights W_i based on $\boldsymbol{\alpha}$, using the ASGQ rule shown in Eqs. (3.92)-(3.96).
4) Calculate $\mathbf{S}$ in Eq. (2.31) and $\tilde{\mathbf{S}}$ in Eq. (2.35) as $\mathbf{D}^{-1}\mathbf{S}_{ref}$, where $\mathbf{DPD}^T = \mathbf{S}_{ref}\mathbf{S}_{ref}^T$. It can be shown $\mathbf{P} = \mathbf{D}^{-1}\mathbf{DPD}^T\mathbf{D}^{-T} = \mathbf{D}^{-1}\mathbf{S}_{ref}\mathbf{S}_{ref}^T\mathbf{D}^{-T} = \left(\mathbf{D}^{-1}\mathbf{S}_{ref}\right)\left(\mathbf{D}^{-1}\mathbf{S}_{ref}\right)^T$.

Note that the ASGQ points $\boldsymbol{\gamma}_i$ and weights W_i that are the quadrature points and weights in Eq. (2.27) for a specific $\boldsymbol{\alpha}$ can be calculated off-line. In addition, when the univariate points and weights are generated by the GHQ rule, the anisotropic sparse-grid quadrature filter is called the anisotropic sparse Gauss-Hermite quadrature filter.

CHAPTER IV

AEROSPACE APPLICATIONS

In this chapter, the proposed filters are applied to three aerospace applications: attitude estimation, orbit determination, and spacecraft relative navigation.

4.1 Spacecraft Attitude Estimation

The spacecraft attitude estimation problem is to estimate the spacecraft‟s orientation from noisy measurement data and known reference observations [90]. Parameters used to represent the three-axis attitude fall into two categories in general: constrained parameters such as the unit quaternion and unconstrained parameters such as the Euler angles, the Rodrigues parameters, the modified Rodrigues parameters (MRPs) [44, 90], and the generalized Rodrigues parameters (GRPs) [50]. The quaternion is widely used to represent the attitude because it is free of singularities and the quaternion kinematics equation is bilinear. However, the unity-norm constraint of the quaternion is often violated in the standard filtering process [50]. A common approach to overcome this problem is to use the quaternion for global nonsingular attitude representation and a set of unconstrained parameters for local attitude representation and filtering process [44, 50]. This approach will be used in the attitude estimator of this chapter.

Many nonlinear filtering methods, such as the Extended Kalman filter (EKF) [42], the Unscented Kalman Filter (UKF) [35, 36], and the Particle Filter (PF) [32, 44] have been employed for spacecraft attitude estimation since it is a nonlinear filtering problem. The EKF is the most widely used nonlinear filtering method for spacecraft attitude

estimation [47]. A simplified Kalman filter and smoother for spacecraft attitude estimation based on the QUEST algorithm was proposed in [48]. A more robust approach named the extended quaternion-estimator, based on the EKF and quadratic constrained programming, was proposed in [49]. Besides the EKF, the UKF [50] demonstrated more accurate and robust performance than the EKF in attitude estimation when the initial attitude estimation error is large. The PF has been shown to achieve better accuracy than the UKF and the EKF at the expense of high computational complexity [43, 44].

In this section, the Sparse Gauss-Hermite Quadrature Filter (SGHQF) is used for the spacecraft attitude estimation. The attitude quaternion kinematics and the gyro and vector observation models are briefly reviewed.

4.1.1 Attitude Kinematics Model

Assume the spacecraft attitude is represented by the quaternion, denoted by $\mathbf{q} = \left[\tilde{\mathbf{q}}^T, q_4\right]^T$, where $\tilde{\mathbf{q}} = \left[\tilde{q}_1, \tilde{q}_2, \tilde{q}_3\right]^T$ is the vector component, the quaternion kinematics can be described by [50]

$$\begin{cases} \dot{\mathbf{q}} = \frac{1}{2} B\left(\mathbf{q}(t)\right)\boldsymbol{\omega}(t) \\ \mathbf{q}^T\mathbf{q} = 1 \end{cases} \tag{4.1}$$

with $\boldsymbol{\omega}$ as the angular velocity, and

$$B(\mathbf{q}) = \begin{bmatrix} q_4\mathbf{I}_{3\times 3} + \left[\tilde{\mathbf{q}}\times\right] \\ -\tilde{\mathbf{q}}^T \end{bmatrix} \tag{4.2}$$

$\left[\tilde{\mathbf{q}}\times\right]$ as the cross-product matrix, i.e. $\left[\tilde{\mathbf{q}}\times\right] = \begin{bmatrix} 0 & -\tilde{q}_3 & \tilde{q}_2 \\ \tilde{q}_3 & 0 & -\tilde{q}_1 \\ -\tilde{q}_2 & \tilde{q}_1 & 0 \end{bmatrix}$ (4.3)

The equivalent discrete-time kinematic equation to propagate the quaternion is

$$\mathbf{q}_{k+1} = \Omega\left(\boldsymbol{\omega}_k\right)\mathbf{q}_k \tag{4.4}$$

where $\mathbf{q}_k = \left[\tilde{\mathbf{q}}_k^T, q_{4k}\right]^T$ is the quaternion at time k and

$$\Omega(\boldsymbol{\omega}_k)=\begin{bmatrix}\cos(0.5\|\boldsymbol{\omega}_k\|\Delta t)\mathbf{I}_{3\times3}-[\boldsymbol{\psi}_k\times] & \boldsymbol{\psi}_k \\ -\boldsymbol{\psi}_k^T & \cos(0.5\|\boldsymbol{\omega}_k\|\Delta t)\end{bmatrix} \tag{4.5}$$

$\boldsymbol{\omega}_k$ is the angular velocity at the *k*-th sampling interval, Δt is the sampling time interval, and $\boldsymbol{\psi}_k=\sin(0.5\|\boldsymbol{\omega}_k\|\Delta t)\boldsymbol{\omega}_k/\|\boldsymbol{\omega}_k\|$ and $[\boldsymbol{\psi}_k\times]$ is a cross product matrix.

In the above attitude kinematics model, the angular velocity is measured by a gyro. A widely used model for the angular velocity measurement is given by [50]

$$\tilde{\boldsymbol{\omega}}(t)=\boldsymbol{\omega}(t)+\boldsymbol{\beta}(t)+\boldsymbol{\eta}_v(t) \tag{4.6}$$

$$\dot{\boldsymbol{\beta}}(t)=\boldsymbol{\eta}_u(t) \tag{4.7}$$

where $\tilde{\boldsymbol{\omega}}(t)$ and $\boldsymbol{\omega}(t)$ are the measured angular velocity and the true angular velocity, respectively, $\boldsymbol{\eta}_v(t)$ and $\boldsymbol{\eta}_u(t)$ are independent Gaussian white noise processes with zero mean and standard deviations of σ_v and σ_u respectively, and $\boldsymbol{\beta}(t)$ is the gyro bias.

In the discrete filter design, the estimated angular velocity is given by [50]

$$\hat{\boldsymbol{\omega}}_{k|k}=\tilde{\boldsymbol{\omega}}_k-\hat{\boldsymbol{\beta}}_{k|k} \tag{4.8}$$

where $\hat{\boldsymbol{\beta}}_{k|k}$ is the updated gyro bias estimate and the propagated gyro bias is given by

$$\hat{\boldsymbol{\beta}}_{k+1|k}=\hat{\boldsymbol{\beta}}_{k|k} \tag{4.9}$$

4.1.2 Vector Observation Model

The vector observation model for attitude estimation is given by [50],

$$\mathbf{y}_k=\boldsymbol{A}_k\mathbf{r}_k+\mathbf{n}_k \tag{4.10}$$

where $\mathbf{n}_k$ is the measurement noise. It is assumed to be white Gaussian noise, with zero mean and covariance $\mathbf{R}_k$. $\mathbf{r}_k$ and $\mathbf{y}_k$ are an observation pair acquired in two different Cartesian coordinate systems at time k, and $\boldsymbol{A}_k$ is the rotation matrix

$$\boldsymbol{A}_k=\boldsymbol{A}(\mathbf{q}_k)=\left[(q_{4k})^2-\tilde{\mathbf{q}}_k^T\tilde{\mathbf{q}}_k\right]\mathbf{I}_{3\times3}+2\tilde{\mathbf{q}}_k\tilde{\mathbf{q}}_k^T-2q_{4k}[\tilde{\mathbf{q}}_k\times] \tag{4.11}$$

4.1.3 Sparse Gauss-Hermite Quadrature Filter for Attitude Estimation

In this section, the SGHQF is used for spacecraft attitude estimation. Following [50], we use the unconstrained generalized Rodrigues parameters (GRPs) to represent the three-component attitude error in the filtering algorithm and use the quaternion to perform attitude propagation. The main difference between the proposed SGHQF in this chapter and the UKF (USQUE) of [50] is the point selection strategy. In addition, the SGHQF and the USQUE account for the effect of process noise on the state error covariance with slightly different techniques. Define a 6×1 state vector as:

$$\mathbf{x} = \begin{bmatrix} \boldsymbol{\delta}\mathbf{p}^T & \boldsymbol{\beta}^T \end{bmatrix}^T \tag{4.12}$$

where $\boldsymbol{\delta}\mathbf{p}$ is the GRPs to represent attitude errors, and is defined by

$$\boldsymbol{\delta}\mathbf{p} = f_c \frac{\boldsymbol{\delta}\tilde{\mathbf{q}}}{a + \delta q_4} \tag{4.13}$$

where a is a parameter in $[0,1]$, $[\boldsymbol{\delta}\tilde{\mathbf{q}}, \delta q_4]$ is the error quaternion, and f_c is a scale factor. When $a = 0$ and $f_c = 1$, Eq. (4.13) gives the Gibbs vector; when $a = f_c = 1$, it gives the standard modified Rodrigues parameters (MRPs).

In the following attitude estimation algorithm, we assume the number of quadrature points is N_p. Given the initial estimates of attitude quaternion $\hat{\mathbf{q}}_{0|0}$, gyro bias $\hat{\boldsymbol{\beta}}_{0|0}$, initial covariance $\mathbf{P}_{0|0}$, initial estimated state vector $\hat{\mathbf{x}}_{0|0} = \begin{bmatrix} \mathbf{0}^T & \hat{\boldsymbol{\beta}}_{0|0}^T \end{bmatrix}^T$, the process noise covariance $\mathbf{Q}_k$, and the measurement covariance $\mathbf{R}_k$, the prediction step and update step of the SGHQF used for attitude estimation can be summarized as follows:

4.1.3.1 Prediction Step

1) Compute the factorization of $\mathbf{P}_{k|k} = \mathbf{S}\mathbf{S}^T$ using the singular value decomposition and set $\boldsymbol{\xi}_{k|k}(i) = \mathbf{S}\boldsymbol{\gamma}_i + \hat{\mathbf{x}}_{k|k}$,

where $$\boldsymbol{\xi}_{k|k}(i) = \begin{bmatrix} \boldsymbol{\xi}_{k|k}^{\delta\hat{p}}(i) \\ \boldsymbol{\xi}_{k|k}^{\hat{\beta}}(i) \end{bmatrix},\ i = 1, \cdots, N_p \tag{4.14}$$

and i is the point index, $\xi_{k|k}^{\delta\hat{p}}(i)$ are attitude error components, and $\xi_{k|k}^{\hat{\beta}}(i)$ are gyro bias components.

Note that $\boldsymbol{\gamma}_1$ is the origin point; $\boldsymbol{\gamma}_i$ $(i=2,\cdots,N_p)$ are the SGHQ points generated by Algorithm I in Section 3.1.1.

Since we use Eq. (4.4) for attitude propagation, we need to transform each of the $\xi_{k|k}^{\delta\hat{p}}(i)$ into an error quaternion and then calculate the corresponding quaternion that will be used in attitude propagation.

Error quaternions $\delta\mathbf{q}_{k|k}^{pre}(i)=\left[\left(\delta\tilde{\mathbf{q}}_{k|k}^{pre}(i)\right)^T,\delta q_{4,k|k}^{pre}(i)\right]^T$ can be calculated by [50]

$$\delta\tilde{\mathbf{q}}_{k|k}^{pre}(i)=f_c^{-1}\left(a+\delta q_{4,k|k}^{pre}\right)\xi_{k|k}^{\delta\hat{p}}(i) \tag{4.15}$$

$$\delta q_{4,k|k}^{pre}(i)=\frac{-a\left|\xi_{k|k}^{\delta\hat{p}}(i)\right|^2+f_c\sqrt{f_c^2+\left(1-a^2\right)\left|\xi_{k|k}^{\delta\hat{p}}(i)\right|^2}}{f_c^2+\left|\xi_{k|k}^{\delta\hat{p}}(i)\right|^2} \tag{4.16}$$

The corresponding quaternions $\hat{\mathbf{q}}_{k|k}^{pre}(i)$ are given by

$$\hat{\mathbf{q}}_{k|k}^{pre}(i)=\delta\mathbf{q}_{k|k}^{pre}(i)\odot\hat{\mathbf{q}}_{k|k} \tag{4.17}$$

The superscript „pre“ represents the prediction step; $\odot$ denotes the quaternion product. Note that $\hat{\mathbf{q}}_{k|k}^{pre}(1)=\hat{\mathbf{q}}_{k|k}$.

2) Predicted quaternions $\hat{\mathbf{q}}_{k+1|k}^{pre}(i)$ are propagated using Eq. (4.4)

$$\hat{\mathbf{q}}_{k+1|k}^{pre}(i)=\Omega\left(\hat{\boldsymbol{\omega}}_k(i)\right)\hat{\mathbf{q}}_{k|k}^{pre}(i) \tag{4.18}$$

where $\hat{\boldsymbol{\omega}}_k(i)=\tilde{\boldsymbol{\omega}}_k-\xi_{k|k}^{\hat{\beta}}(i)$.

Since the state vector is defined by GRPs, quaternions $\hat{\mathbf{q}}_{k+1|k}^{pre}(i)$ need to be transformed into GRPs in order to use the SGHQF algorithm. First, the corresponding error quaternions $\delta\mathbf{q}_{k+1|k}^{pre}(i)=\left[\left(\delta\tilde{\mathbf{q}}_{k+1|k}^{pre}(i)\right)^T,\delta q_{4,k+1|k}^{pre}(i)\right]$ are calculated by $\delta\mathbf{q}_{k+1|k}^{pre}(i)=\hat{\mathbf{q}}_{k+1|k}^{pre}(i)\odot\left[\hat{\mathbf{q}}_{k+1|k}^{pre}(1)\right]^{-1}$. Then, the corresponding predicted attitude error components $\xi_{k+1|k}^{\delta\hat{p}}(i)$ and gyro-bias components $\xi_{k+1|k}^{\hat{\beta}}(i)$ of $\xi_{k+1|k}(i)$ are calculated from the error quaternions by

$$\xi_{k+1|k}^{\delta\hat{p}}(i) = f_c \frac{\delta\tilde{\mathbf{q}}_{k+1|k}^{pre}(i)}{a + \delta q_{4,k+1|k}^{pre}(i)} \tag{4.19}$$

and

$$\xi_{k+1|k}^{\hat{\beta}}(i) = \xi_{k|k}^{\hat{\beta}}(i) \tag{4.20}$$

3) The corresponding propagated state vector value and covariance are given by the SGHQF algorithm

$$\hat{\mathbf{x}}_{k+1|k} = \sum_{i=1}^{N_p} w_i \xi_{k+1|k}(i) \tag{4.21}$$

$$\mathbf{P}_{k+1|k} = \sum_{i=1}^{N_p} w_i \left(\xi_{k+1|k}(i) - \hat{\mathbf{x}}_{k+1|k}\right)\left(\xi_{k+1|k}(i) - \hat{\mathbf{x}}_{k+1|k}\right)^T + \mathbf{Q}_k \tag{4.22}$$

where $\mathbf{Q}_k = \begin{bmatrix} \left(\sigma_v^2 \Delta t + \frac{1}{3}\sigma_u^2 \Delta t^3\right) I_{3\times3} & -\left(\frac{1}{2}\sigma_u^2 \Delta t^2\right) I_{3\times3} \\ -\left(\frac{1}{2}\sigma_u^2 \Delta t^2\right) I_{3\times3} & \left(\sigma_u^2 \Delta t\right) I_{3\times3} \end{bmatrix}$ with Δt as the measurement sampling period.

4) Transform the first three elements of $\hat{\mathbf{x}}_{k+1|k}$, i.e. $\delta\hat{\mathbf{p}}_{k+1|k}$ into the error quaternion form and then calculate the predicted quaternion $\hat{\mathbf{q}}_{k+1|k}$. The corresponding error quaternion $\delta\mathbf{q}_{k+1|k} = \left[\delta\tilde{\mathbf{q}}_{k+1|k}^T, \delta q_{4,k+1|k}\right]^T$ is given by

$$\delta\tilde{\mathbf{q}}_{k+1|k} = f_c^{-1}\left(a + \delta q_{4,k+1|k}\right)\delta\hat{\mathbf{p}}_{k+1|k} \tag{4.23}$$

and

$$\delta q_{4,k+1|k} = \frac{-a\left|\delta\hat{\mathbf{p}}_{k+1|k}\right|^2 + f_c\sqrt{f_c^2 + \left(1-a^2\right)\left|\delta\hat{\mathbf{p}}_{k+1|k}\right|^2}}{f_c^2 + \left|\delta\hat{\mathbf{p}}_{k+1|k}\right|^2} \tag{4.24}$$

Then, the predicted quaternion $\hat{\mathbf{q}}_{k+1|k}$ is given by

$$\hat{\mathbf{q}}_{k+1|k} = \delta\mathbf{q}_{k+1|k} \odot \hat{\mathbf{q}}_{k+1|k}^{pre}(1) \tag{4.25}$$

Reset the first three elements of $\hat{\mathbf{x}}_{k+1|k}$ to zeros. This step is used to move information from one part of the estimate to another part [50].

4.1.3.2 Update Step

1) Compute the factorization $\mathbf{P}_{k+1|k} = \tilde{\mathbf{S}}\tilde{\mathbf{S}}^T$ and set $\boldsymbol{\delta}_{k+1|k}(i) = \tilde{\mathbf{S}}\boldsymbol{\gamma}_i + \hat{\mathbf{x}}_{k+1|k}$.

where $\boldsymbol{\delta}_{k+1|k}(i) = \left[\boldsymbol{\delta}^{\delta\hat{p}}_{k+1|k}(i)^T \quad \boldsymbol{\delta}^{\hat{\beta}}_{k+1|k}(i)^T\right]^T$ (4.26)

Since GRPs cannot be directly used in the measurement equation as given by (4.10), $\boldsymbol{\delta}^{\delta\hat{p}}_{k+1|k}(i)$ must be transformed into the error quaternion form. Then we can calculate the corresponding quaternions $\hat{\mathbf{q}}^{upd}_{k+1|k}(i)$ which will be used in Eq. (4.10).

Note that we use $\hat{\mathbf{q}}^{upd}_{k+1|k}$ to denote the quaternion used in the update step to avoid confusion with the quaternion $\hat{\mathbf{q}}^{pre}_{k+1|k}$ used in the prediction step. Error quaternions $\boldsymbol{\delta}\mathbf{q}^{upd}_{k+1|k}(i)$ are given by

$$\boldsymbol{\delta}\mathbf{q}^{upd}_{k+1|k}(i) = \left[\left(\boldsymbol{\delta}\tilde{\mathbf{q}}^{upd}_{k+1|k}(i)\right)^T, \delta q^{upd}_{4,k+1|k}(i)\right]^T \tag{4.27}$$

where
$$\boldsymbol{\delta}\tilde{\mathbf{q}}^{upd}_{k+1|k}(i) = f_c^{-1}\left(a + \delta q^{upd}_{4,k+1|k}\right)\boldsymbol{\delta}^{\delta\hat{p}}_{k+1|k}(i) \tag{4.28}$$

$$\delta q^{upd}_{4,k+1|k}(i) = \frac{-a\left|\boldsymbol{\delta}^{\delta\hat{p}}_{k+1|k}(i)\right|^2 + f_c\sqrt{f_c^2 + \left(1-a^2\right)\left|\boldsymbol{\delta}^{\delta\hat{p}}_{k+1|k}(i)\right|^2}}{f_c^2 + \left|\boldsymbol{\delta}^{\delta\hat{p}}_{k+1|k}(i)\right|^2} \tag{4.29}$$

Then, quaternions $\hat{\mathbf{q}}^{upd}_{k+1|k}(i)$ are calculated by

$$\hat{\mathbf{q}}^{upd}_{k+1|k}(i) = \boldsymbol{\delta}\mathbf{q}^{upd}_{k+1|k}(i) \odot \hat{\mathbf{q}}_{k+1|k} \tag{4.30}$$

2) States are updated using the SGHQF algorithm as follows

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{L}_{k+1}\left(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}\right) \tag{4.31}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{L}_{k+1}\mathbf{P}^T_{xy} \tag{4.32}$$

where $\mathbf{y}_{k+1}$ is the true measurement value at time $k+1$.

$$\hat{\mathbf{y}}_{k+1} = \sum_{i=1}^{N_p} w_i \boldsymbol{A}\left(\hat{\mathbf{q}}^{upd}_{k+1|k}(i)\right)\mathbf{r}_{k+1} \tag{4.33}$$

$$\mathbf{P}_{xy} = \sum_{i=1}^{N_p} w_i\left(\boldsymbol{\delta}_{k+1|k}(i) - \hat{\mathbf{x}}_{k+1|k}\right)\left(A\left(\hat{\mathbf{q}}^{upd}_{k+1|k}(i)\right)\mathbf{r}_{k+1} - \hat{\mathbf{y}}_{k+1}\right)^T \tag{4.34}$$

$$\mathbf{P}_{yy} = \sum_{i=1}^{N_p} w_i\left(A\left(\hat{\mathbf{q}}^{upd}_{k+1|k}(i)\right)\mathbf{r}_{k+1} - \hat{\mathbf{y}}_{k+1}\right)\left(A\left(\hat{\mathbf{q}}^{upd}_{k+1|k}(i)\right)\mathbf{r}_{k+1} - \hat{\mathbf{y}}_{k+1}\right)^T \tag{4.35}$$

$$\mathbf{L}_k = \mathbf{P}_{xy}\left(\mathbf{R}_k + \mathbf{P}_{yy}\right)^{-1} \tag{4.36}$$

3) The updated quaternion $\hat{\mathbf{q}}_{k+1|k+1}$ is then calculated using the first three elements of $\hat{\mathbf{x}}_{k+1|k+1}$, i.e. $\boldsymbol{\delta}\hat{\mathbf{p}}_{k+1|k+1}$ and $\hat{\mathbf{q}}_{k+1|k}$, i.e. $\hat{\mathbf{q}}_{k+1|k+1} = \boldsymbol{\delta}\mathbf{q}_{k+1|k+1} \odot \hat{\mathbf{q}}_{k+1|k}$ where $\boldsymbol{\delta}\mathbf{q}_{k+1|k+1} = \left[\boldsymbol{\delta}\tilde{\mathbf{q}}^T_{k+1|k+1}, \delta q_{4,k+1|k+1}\right]^T$ with

$$\boldsymbol{\delta}\tilde{\mathbf{q}}_{k+1|k+1} = f_c^{-1}\left(a + \delta q_{4,k+1|k+1}\right)\boldsymbol{\delta}\hat{\mathbf{p}}_{k+1|k+1} \tag{4.37}$$

$$\boldsymbol{\delta}\mathbf{q}_{4,k+1|k+1} = \frac{-a\left|\boldsymbol{\delta}\hat{\mathbf{p}}_{k+1|k+1}\right|^2 + f_c\sqrt{f_c^2 + \left(1-a^2\right)\left|\boldsymbol{\delta}\hat{\mathbf{p}}_{k+1|k+1}\right|^2}}{f_c^2 + \left|\boldsymbol{\delta}\hat{\mathbf{p}}_{k+1|k+1}\right|^2} \tag{4.38}$$

Reset $\boldsymbol{\delta}\hat{\mathbf{p}}_{k+1|k+1}$ to zeros.

The SGHQF for the spacecraft attitude estimation can be summarized in Algorithm II.

Algorithm 4.1 SGHQF for Spacecraft Attitude Estimation

$[\hat{\mathbf{q}}, \mathbf{P}] = \text{SGHQF\_ATTITUDE}[\hat{\mathbf{q}}_{0|0}, \mathbf{P}_{0|0}, \hat{\mathbf{x}}_{0|0}, T, \mathbf{y}, \Delta t, n, L]$

($\hat{\mathbf{q}}_{0|0}$: initial estimated quaternion; $\mathbf{P}_{0|0}$: initial covariance matrix; T : total time of the simulation; $\mathbf{y}$: measurement values; Δt : propagation step size; n : dimension of the filtering algorithm; L : accuracy level for the SGHQ; $\hat{\mathbf{q}}$: estimated quaternions ($\hat{\mathbf{q}}_{k|k} \in \hat{\mathbf{q}}, k = 0, \cdots, T/\Delta t$); $\mathbf{P}$: covariance matrix ($\mathbf{P}_{k|k} \in \mathbf{P}, k = 0, \cdots, T/\Delta t$))

Generate SGHQ points and weights $[\chi, W] = \text{SGHQ}[n, L]$ using Algorithm I;

For k=0: T/ Δt

Prediction:

1) Factorize $\mathbf{P}_{k|k} = \mathbf{S}\mathbf{S}^T$ and let $\xi_{k|k}(i) = \mathbf{S}\gamma_i + \hat{\mathbf{x}}_{k|k}$ ($\gamma_i \in \chi$).

2) Calculate quaternions $\hat{\mathbf{q}}_{k|k}^{pre}$ by Eqs.(4.15)-(4.17).

3) Propagate quaternions $\hat{\mathbf{q}}_{k|k}^{pre}$ to $\hat{\mathbf{q}}_{k+1|k}^{pre}$ by Eq. (4.18).

4) Calculate attitude errors $\xi_{k+1|k}^{\delta\hat{p}}(i)$ by Eq. (4.19).

5) Calculate the predicted state $\hat{\mathbf{x}}_{k+1|k}$ and covariance $\mathbf{P}_{k+1|k}$ using Eqs. (4.21) and (4.22).

6) Calculate the predicted quaternion $\hat{\mathbf{q}}_{k+1|k}$ by Eqs. (4.23)-(4.25) and reset the first three elements of $\hat{\mathbf{x}}_{k+1|k}$ to zeros.

Algorithm 4.1 (continued)

Update:

1) Factorize $\mathbf{P}_{k+1|k} = \tilde{\mathbf{S}}\tilde{\mathbf{S}}^T$ and recalculate points $\boldsymbol{\delta}_{k+1|k}(i) = \tilde{\mathbf{S}}\boldsymbol{\gamma}_i + \hat{\mathbf{x}}_{k+1|k}$.

2) Calculate quaternions $\hat{\mathbf{q}}^{upd}_{k+1|k}(i)$ by Eqs. (4.27)-(4.30).

3) Update the state $\hat{\mathbf{x}}_{k+1|k+1}$ and covariance $\mathbf{P}_{k+1|k+1}$ by Eqs. (4.31)-(4.36); output $\mathbf{P}_{k+1|k+1}$ to $\mathbf{P}$.

4) Calculate the error quaternion $\delta\mathbf{q}_{k+1|k+1}$ using Eqs. (4.37), (4.38) and update the quaternion $\hat{\mathbf{q}}_{k+1|k+1}$ by $\hat{\mathbf{q}}_{k+1|k+1} = \boldsymbol{\delta}\mathbf{q}_{k+1|k+1} \odot \hat{\mathbf{q}}_{k+1|k}$; output $\hat{\mathbf{q}}_{k+1|k+1}$ to $\hat{\mathbf{q}}$. Reset the first three elements of $\hat{\mathbf{x}}_{k+1|k+1}$ to zeros.

END FOR

4.1.4 Numerical Results and Analysis

Both the sparse Gauss-Hermite quadrature filter (SGHQF) and anisotropic sparse Gauss-Hermite quadrature filter (ASGHQF) are used.

4.1.4.1 The Performance of SGHQF

In this section, several test cases are simulated to evaluate the performance of SGHQF against several well-known nonlinear filtering techniques. The simulation of the true attitude and sensor data is taken from [50]. The orbit parameters used here are obtained from the TRMM spacecraft [50]. Only the three-axis magnetometer (TAM) and gyroscopes are used for measurements. The magnetic field reference model is the 10th International Geomagnetic Reference Field Model. The noise of the TAM model is zero mean white Gaussian noise with a standard deviation of 50 nT. The gyro noise is also assumed to have a white Gaussian distribution with zero mean and standard deviation of

$\sigma_u = 3.1623\times10^{-10}\,\text{rad/s}^{3/2}$ and $\sigma_v = 3.1623\times10^{-7}\,\text{rad/s}^{1/2}$ [3]. The initial gyro bias is assumed to be zero. The parameters in GRPs are set to $a = 1$ and $f_c = 4$.

Since we have shown that the CKF and UKF using certain parameters are special cases of the SGHQF, it is necessary to conduct a comprehensive comparison of these nonlinear filters, which include EKF, CKF, UKF, and the GHQF.

As discussed before, different numbers of points for the univariate GHQ with different accuracy levels can be used for SGHQ. In the following cases, we use three point-selection strategies: L, $2L-1$, and 2^L-1, for the level-L univariate GHQ . For convenience, we denote them the 1st SGHQF, 2nd SGHQF, and 3rd SGHQF respectively. The total number of SGHQ points with the accuracy level-2 and level-3 using these three point-selection strategies for 6-dimensional problems are listed in Table 4.1.

Table 4.1 Total number of points for 6-dimensional SGHQFs (level-2 and level-3), CKF, UKF, and GHQF

	level-2	level-3
1st SGHQF	13	85
2nd SGHQF	13	97
3rd SGHQF	13	109
UKF	13	
CKF	12	
GHQF (m_L=3)	729	

The following simulation results are all averaged values of 50 Monte-Carlo runs.

Case 1: No Initial Estimation Error And Zero Initial Bias Estimate

In the first case (Case 1), EKF, SGHQFs (level-2 and level-3), UKF, CKF and the conventional GHQF (level-2) using 3 points for each dimension are compared assuming that there is no initial attitude estimation error and the initial bias estimate is set to zero. The initial attitude covariance is set to $\left(0.5^{\circ}\right)^2$ (0.5° is the rotation angle about the eigen-axis of rotation) and the bias covariance is set to $\left(0.1^{\circ}/\mathrm{h}\right)^2$. Since the simulation exhibits very similar performance for all filters in this case, the results are not shown here.

Case 2: 30 degree Initial Attitude Estimation Error And Zero Initial Bias Estimate

The second case (Case 2) is to add 30 degree error to the initial attitude estimate using the same parameters as those in Case 1. The initial bias estimate is still set to zeros. The initial attitude covariance is set to $\left(30^{\circ}\right)^2$ and the bias covariance is set to $\left(0.1^{\circ}/\mathrm{h}\right)^2$. We use the same initial covariance for the EKF, UKF, CKF, and SGHQFs.

The norm of attitude estimation error for this case is shown in Figs. 4.1-4.2. The results of SGHQFs with different accuracy levels and different univariate point-selection strategies are shown in Figure. 4.1. The 2nd SGHQF (level-2) exhibits identical performance with the 3rd SGHQF (level-2) since they use the same point set in the filtering algorithm. In addition, they both have slightly better performance than the 1st SGHQF (level-2). Furthermore, the SGHQFs (level-3) using different point-selection strategies show no noticeable difference (they are overlapped and indistinguishable in the figure). In Figure. 4.2, SGHQFs are compared with EKF, UKF, CKF, and the conventional GHQF. Because SGHQFs at level-3 with different point-selection strategies have very close performance, we use the SGHQF (level-3) to denote all of them. It can be seen that both the SGHQFs (level-2) and the SGHQF (level-3) are more accurate than the EKF. Furthermore, the SGHQF (level-3), the conventional GHQF, and the UKF ($\kappa = 0$) /CKF exhibit nearly the same performance (they are overlapped and indistinguishable).

They are slightly better than the 2nd (or the 3rd) SGHQF (level-2) and the 1st SGHQF (level-2). Different SGHQFs at level-2 show slightly different performance while they all work well for this case and are much better than EKF.

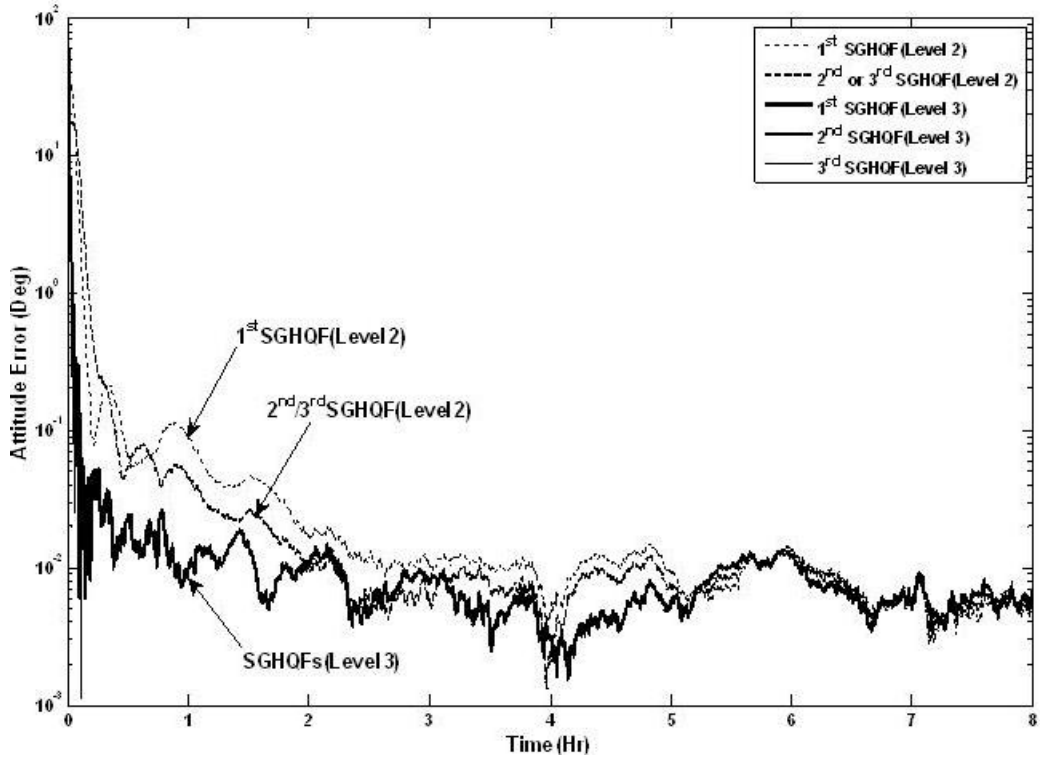

Figure 4.1 Norm of attitude estimation errors for SGHQFs using different point-selection strategies and accuracy levels (Case 2)

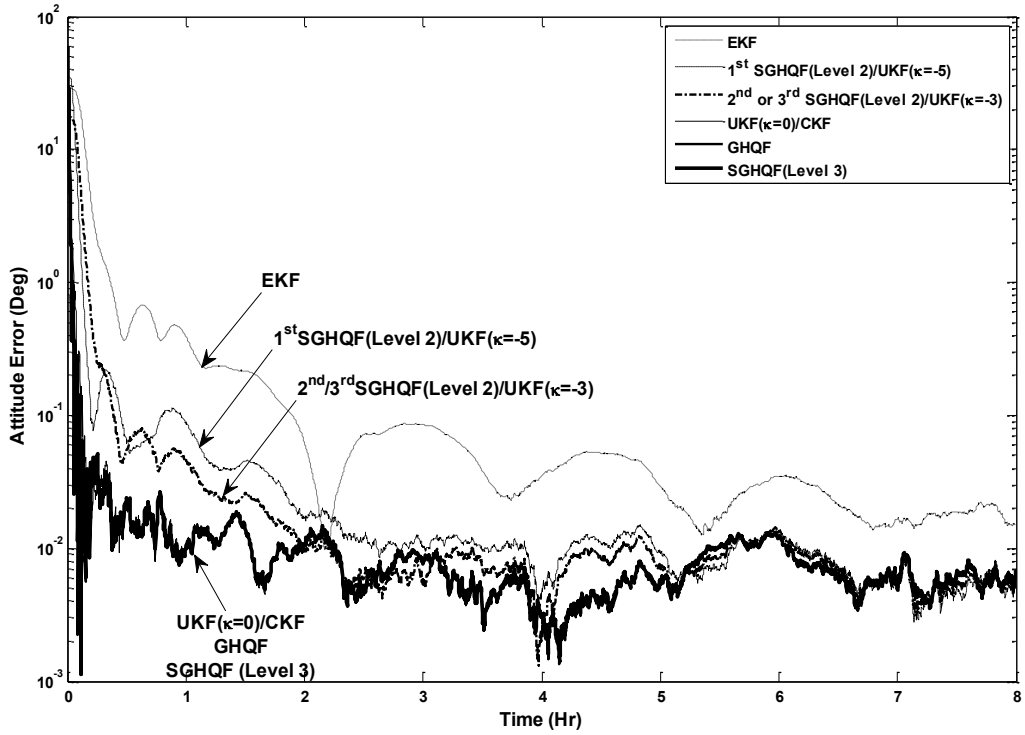

Figure 4.2 Norm of attitude estimation errors in comparison with EKF, UKF, CKF and GHQF (Case 2)

Case 3: 30 Degree Initial Attitude Estimation Error And $10^{\circ}/\mathrm{h}$ Initial Gyro Bias Estimate In The y-axis

The third simulation case (Case 3) adds $10^{\circ}/\mathrm{h}$ initial gyro bias in the y-axis in addition to the 30 degree initial attitude estimation error. The initial attitude covariance is set to $\left(30^{\circ}\right)^2$ and the initial bias covariance is set to $\left(10^{\circ}/\mathrm{h}\right)^2$.

In Figures 4.3-4.5, the performance of SGHQFs with different accuracy levels and different point-selection strategies, UKF, CKF, the conventional GHQF (m_L=3), and EKF are compared. In Figure 4.3, CKF, SGHQFs (level-2) and EKF are compared. It is shown that only CKF converges within the 3σ error bounds, while EKF, the 1st SGHQF, and the 2nd or 3rd SGHQF (level-2) do not. In addition, the 1st SGHQF (level-2) converges faster than the 2nd or 3rd SGHQF (level-2). Furthermore, all SGHQFs (level-2) converge

faster than the EKF. Note that by *Theorem 3.2*, the 1st SGHQF (level-2) and the 2nd or the 3rd SGHQF (level-2) have identical performance with the UKF using the parameters $\kappa = -5$ and $\kappa = -3$, respectively. In Figure 4.4, it is shown that the SGHQF (level-3) converges into the 3σ error bound, within 1 hour, while the CKF needs more than three hours. Moreover, both SGHQF (level3) and CKF have better performance than EKF and SGHQF (level-2). The UKF with the suggested parameter ($\kappa = 3 - n$) does not converge into 3σ error bound within 8 hours. The result of the GHQF is not shown in Figures 4.3 and 4.4 because the performance is very close to SGHQFs (level-3). It is shown in Figure 4.6.

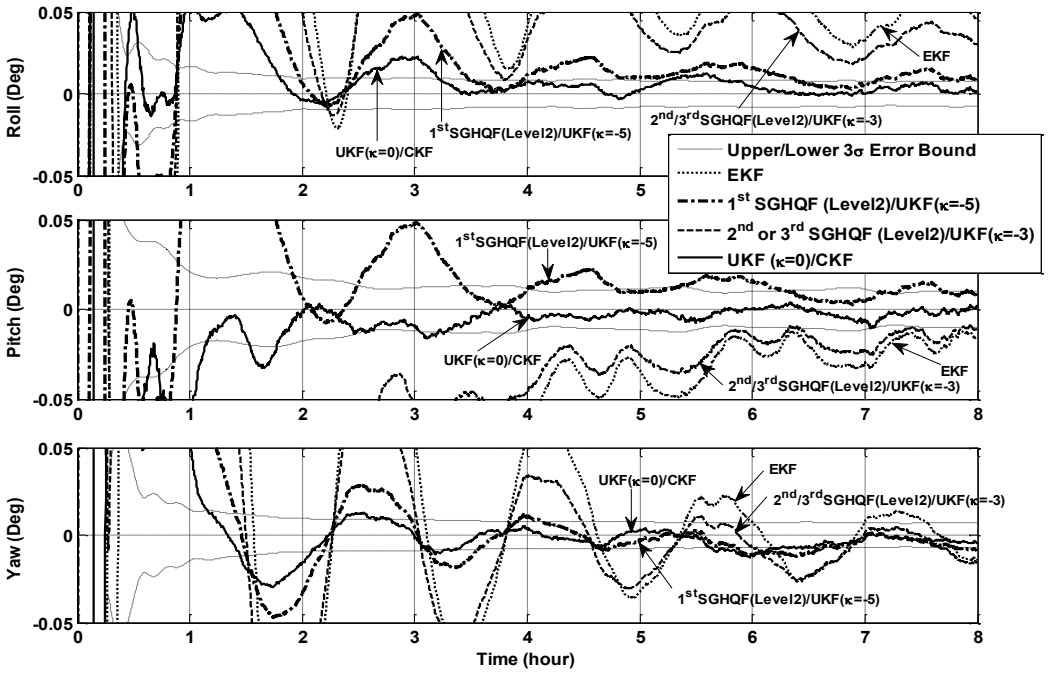

Figure 4.3 Attitude errors of SGHQFs (level-2), UKF, CKF, and EKF with 3σ error bounds (Case 3)

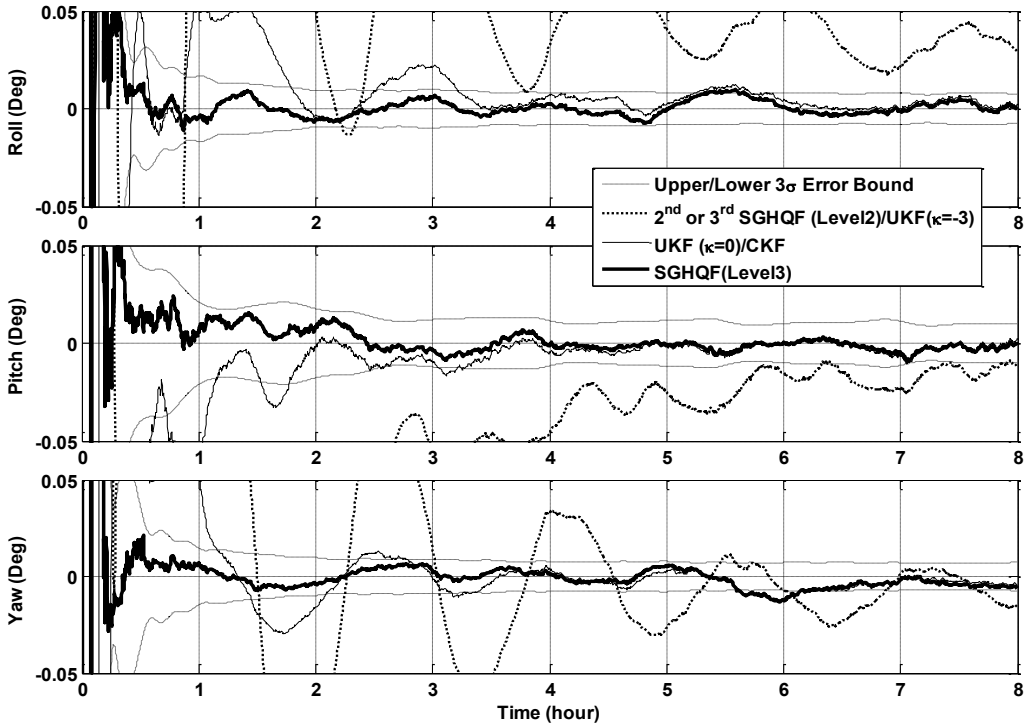

Figure 4.4 Attitude errors of SGHQFs (level3), UKF, and CKF with 3σ error bounds (Case 3)

The norm of total estimation errors using the SGHQFs with different accuracy levels and different point-selection strategies are shown in Figure 4.5. In this case, the 2nd SGHQF (level-2) and the 3rd SGHQF (level-2) (they are identical) have worse performance than the 1st SGHQF (level-2), which is also shown in Figure 4.4. The SGHQFs with the accuracy level-3 using different point-selection strategies still show no noticeable difference. The performance comparison of EKF, GHQF (m_L=3), UKF, CKF, and SGHQFs (level-2 and level-3) is shown in Figure 4.6. The SGHQF (level-3) is still used to represent all SGHQFs (level-3) using different point-selection strategies, since the difference between them is nearly indistinguishable. It can be seen that the 2nd (or the 3rd) SGHQF (level-2) has identical performance with the UKF ($\kappa = -3$) and the 1st SGHQF (level-2) has identical performance with the UKF ($\kappa = -5$), as predicted by Theorem 3.2.

Moreover, different SGHQFs with the accuracy level-2 exhibit more obvious difference in this case. The SGHQF (level-3) has better performance than all SGHQFs with the accuracy level-2, the UKFs, and the CKF. It implies that the SGHQF (level-3) is able to capture large uncertainties more efficiently than other filters.

The conventional GHQF (m_L=3) exhibits performance that is similar to the SGHQF (level-3). However, considering the computational burden, the SGHQF is superior to the conventional GHQF. In addition, EKF shows the worst performance in this case.

For this case of study, it can be noticed that the univariate point-set selection strategy may greatly affect the performance of the SGHQF with the accuracy level-2, which can be considered as a tunable parameter like the κ parameter for the UKF. However, the SGHQF with the accuracy level-3 is not sensitive to the univariate point-set selection. In this sense, the SGHQF will reach a more stable and guaranteed estimation performance as the accuracy level increases. Recall that the UKF can only ensure the accuracy level up to 3rd order polynomials while the SGHQF with the accuracy level-3 is accurate up to 5th order polynomials. In addition, from the filter design perspective, the SGHQF algorithm is very flexible with regard to the choice of accuracy levels, depending on the estimation requirements and computational resources. The designer can easily use a higher accuracy level (L>3) SGHQF to solve more complex multi-dimensional nonlinear estimation problems.

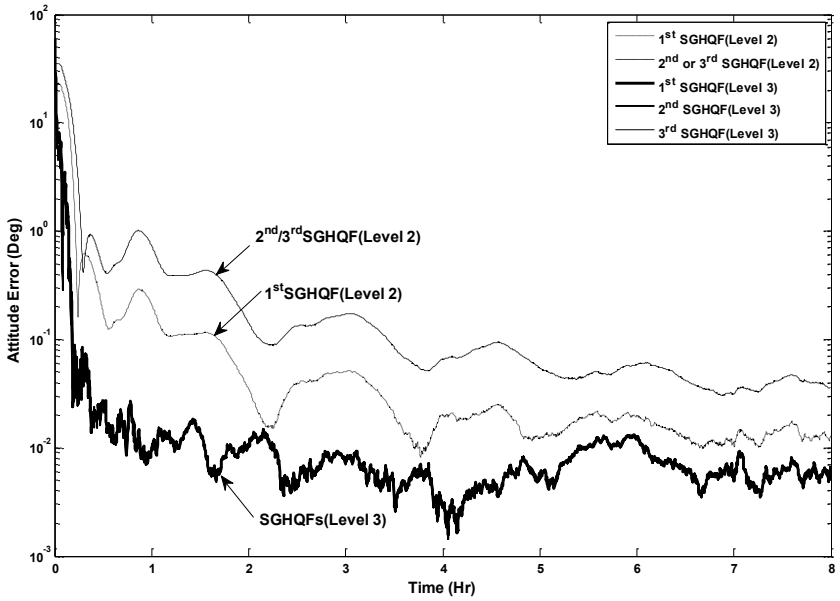

Figure 4.5 Norm of attitude estimation errors for SGHQFs using different point-selection strategies and accuracy levels (Case 3)

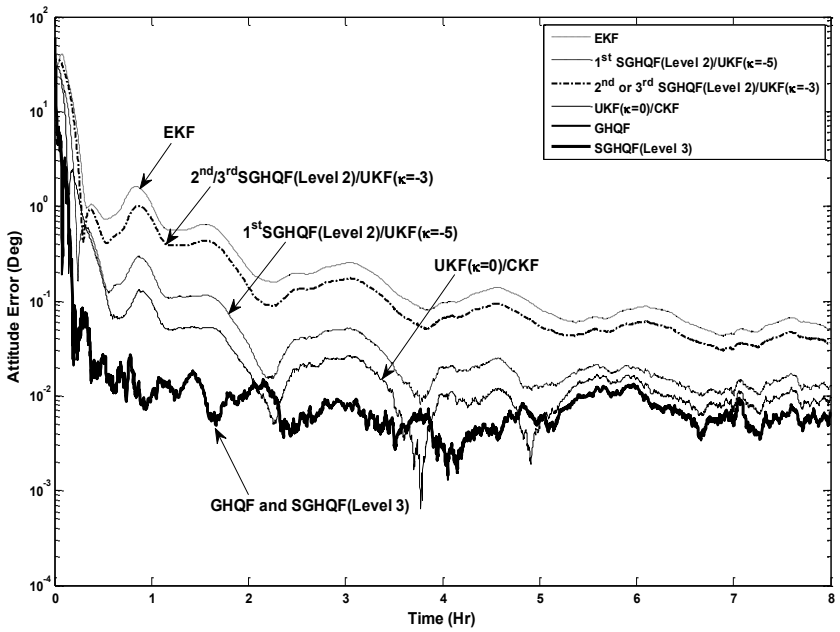

Figure 4.6 Norm of attitude estimation errors (Case 3)

4.1.4.2 The Performance of ASGHQF

In this section, the performance of the ASGHQF is demonstrated and compared with the EKF, UKF, CKF, SGHQF, and the GHQF.

The initial true gyro bias is assumed to be 0.1° / hour . The initial attitude estimation errors and the initial gyro bias estimation errors are assumed to be random and follow a normal distribution with a standard deviation of 30° and 10° / hour , respectively.

Considering the different scales of the GRPs and the gyro bias, the scaling matrix in Section 3.3.3 is chosen to be **D**=diag([1/3600, 1/3600, 1/3600,1,1,1]). The importance of the vector $\boldsymbol{\alpha}$ and the numbers of points of different filters used in the simulation are shown in Table 4.2. For comparison, six cases of $\boldsymbol{\alpha}$ are considered. The values of $\boldsymbol{\alpha}$ are chosen such that more points are placed in the dimensions with larger uncertainties. Note that by Eq. (3.74), the p^{th} ASGHQF ($L^{\alpha}=3$) generates more points in the first p ($p=1,\cdots,6$) dimensions since these dimensions are more important due to larger uncertainties.

Table 4.2 Importance vector $\boldsymbol{\alpha}$ of ASGHQF and numbers of points of different filters

	Importance vector	Number of points
ASGHQF($L^{\alpha}=2$)/SGHQF(L=2) /UKF($\kappa=3-n$)	$\boldsymbol{\alpha}=[1,1,1,1,1,1]$	13
1st ASGHQF($L^{\alpha}=3$)	$\boldsymbol{\alpha}=[1,2,2,2,2,2]$	15
2nd ASGHQF($L^{\alpha}=3$)	$\boldsymbol{\alpha}=[1,1,2,2,2,2]$	25
3rd ASGHQF($L^{\alpha}=3$)	$\boldsymbol{\alpha}=[1,1,1,2,2,2]$	37
4th ASGHQF($L^{\alpha}=3$)	$\boldsymbol{\alpha}=[1,1,1,1,2,2]$	53
5th ASGHQF($L^{\alpha}=3$)	$\boldsymbol{\alpha}=[1,1,1,1,1,2]$	73
6th ASGHQF($L^{\alpha}=3$)/SGHQF (L=3)	$\boldsymbol{\alpha}=[1,1,1,1,1,1]$	97
GHQF		729

Recall that the 6th ASGHQF ($L^{\alpha}=3$) is identical to SGHQF (L=3) as discussed in Section 3.3, since all the elements of $\boldsymbol{\alpha}$ are the same. The points and weights of the SGHQs and the ASGHQs are obtained using the SGHQ algorithm of Ref. [25]and Eq. (3.74), respectively. Moreover, accuracy levels 2 and 3 (i.e. L=2 or 3) are used for the SGHQF and accuracy level-3 is used for the ASGHQF (i.e. $L^{\alpha}=3$). Note that the level-2 ($L^{\alpha}=2$) ASGHQ is the same as the level-2 SGHQ, according to *Remark 3.9*, and the UKF ($\kappa=3-n$) is also the same as the level-2 SGHQF [25].

In Figure 4.7-4.9, the performance of ASGHQFs with different $\boldsymbol{\alpha}$, EKF, UKF ($\kappa=3-n$), CKF, GHQF, and SGHQF are compared based on 100 Monte Carlo runs . In Figure 4.7, the attitude errors of the EKF, UKF, CKF, and GHQF are compared. The 3σ bound of the GHQF is used, since it has the highest accuracy. It is shown that in the simulated period of time, the EKF and the UKF ($\kappa=3-n$) do not converge into the 3σ

bound, whereas the CKF and GHQF do. In Figure 4.8, the EKF, the ASGHQFs with different $\boldsymbol{\alpha}$, and the GHQF are compared. Because the 5th ASGHQF and the 6th ASGHQF have very close performance to the GHQF, they are not shown. In Figure 4.9, the norms of the total attitude estimation errors are compared for the EKF, UKF, CKF, ASGHQFs and GHQF. It is can be seen that the EKF is the worst and the CKF is more accurate than the UKF ($\kappa = 3 - n$). In addition, all ASGHQFs are more accurate than the UKF, as predicted by *Theorem 3.5*, since the UKF is in fact a level-2 SGHQF. The 3rd ASGHQF and the 4th ASGHQF are both more accurate than the CKF. The 5th ASGHQF, the 6th ASGHQF, and the GHQF are very close and are nearly indistinguishable from each other.

As for the computational efficiency of the filtering algorithm, it is worth noting that all point-based Gaussian approximation filters share the same filtering algorithm given by Eqs.(2.29)-(2.35). The difference lies in the methods of computing the points $\boldsymbol{\gamma}_i$ and weights W_i. Note that they are generated offline. The more points there are, the more function evaluations and matrix-vector computations will be involved in the online filtering algorithm as seen from Eqs. (2.29)-(2.35). Therefore, the main computational load depends on the number of points. The other slight difference of ASGHQF from others is on the scaling of the covariance matrix and the new formulae for computing $\mathbf{S}$ in Eq. (2.31) and $\tilde{\mathbf{S}}$ in Eq. (2.35) (Section 2.2.1.2). However, the extra computation only involves two matrix multiplications, i.e. $\mathbf{DPD}^T$, which is very minor compared to the rest of the computation load. The computation load of the new ASGHQF lies in between UKF and GHQF/SGHQF. But, it is more accurate than UKF and maintains close performance to GHQF and SGHQF. The primary advantage of the ASGHQF is that it is

more flexible to use than the SGHQF and GHQF, allowing the designer to tune the importance vector $\boldsymbol{\alpha}$ to balance the computational load with the accuracy requirement.

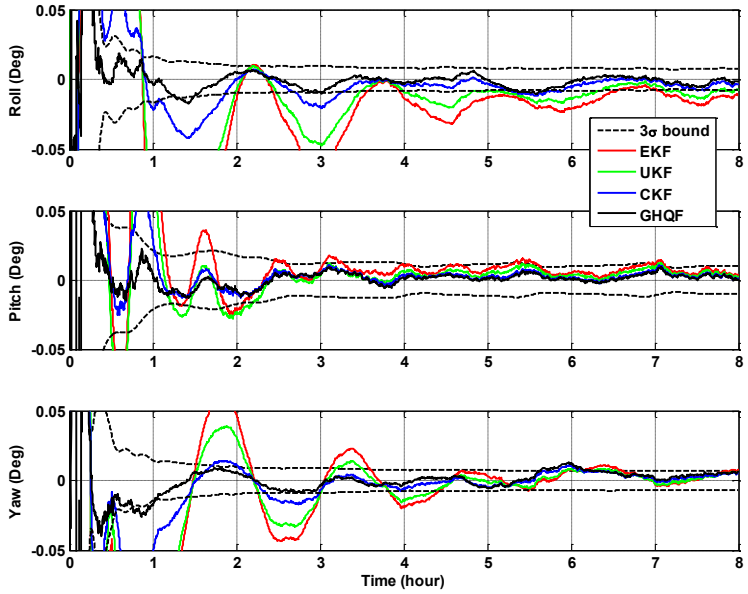

Figure 4.7 Attitude estimation errors of EKF, UKF, CKF, and GHQF

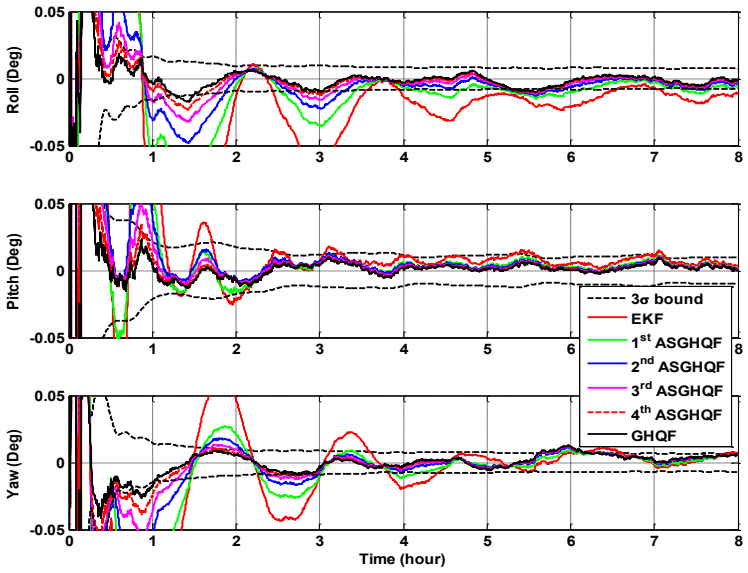

Figure 4.8 Attitude estimation errors of EKF, ASGHQFs, and GHQF

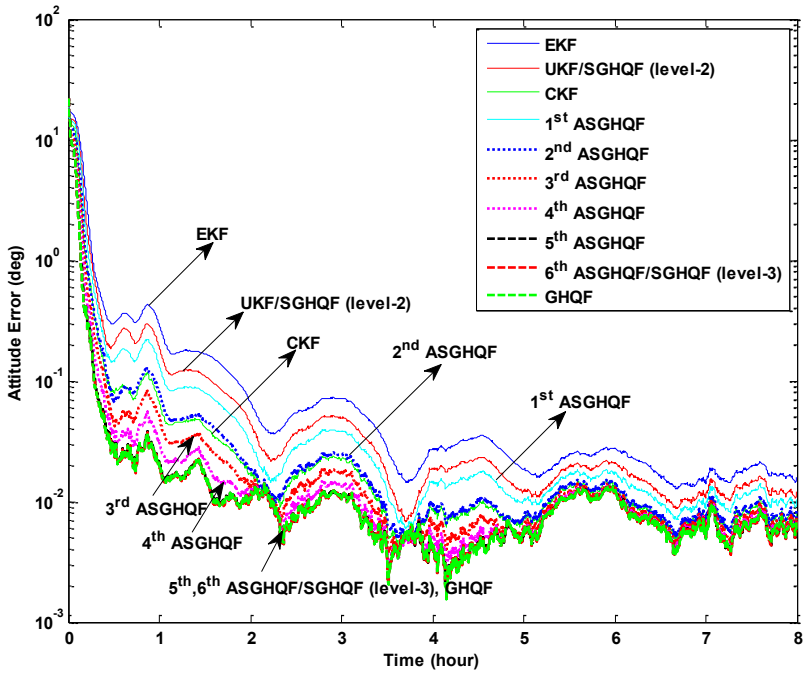

Figure 4.9 Comparison of total attitude estimation error norms using EKF, UKF, CKF, ASGHQFs and GHQF

4.2 Spacecraft Orbit Determination

In this section, the performance of the SGQF is demonstrated by the orbit determination problem and is compared with EKF, UKF, CKF, and GHQF. The orbit determination problem is to obtain accurate satellite position and velocity from noisy observations.

The low-earth orbit (LEO) satellite dynamics can be described by [91]

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_G + \mathbf{a}_D + \boldsymbol{v} \tag{4.39}$$

where $\mathbf{r} = [x, y, z]^T$ is the position of the satellite in the inertial coordinate frame (***I-J-K***), $r = \sqrt{x^2 + y^2 + z^2}$, $\boldsymbol{v}$ is a white Gaussian process noise, and $\mathbf{a}_G$ is the instantaneous acceleration due to the J_2 perturbation [92]. J_2 is the dimensionless second zonal harmonic that quantifies the major oblateness effect of the Earth. The J_2 perturbation

may cause a noticeable precession of the LEO satellite orbits. $\mathbf{a}_D$ is the atmospheric drag with

$$\mathbf{a}_D = -\frac{1}{2}\frac{C_d A}{m_s}\rho_d v_{rel}\mathbf{v}_{rel} \tag{4.40}$$

where ρ_d is the atmospheric density; m_s is the satellite"s mass; C_d is the drag coefficient; $\mathbf{v}_{rel}$ is the velocity vector relative to the rotating atmosphere and A is cross-sectional area [51]. In this chapter, we only consider J_2 and atmospheric drag perturbations. The lunar/solar gravity perturbation is very small and is negligible for the low-earth orbit (LEO) satellite. All the unmodeled perturbations are regarded as the process noise.

The measurement model is described by

$$\begin{cases} az = \tan^{-1}\left(\dfrac{\rho_e}{\rho_n}\right) + n_{az} \\ el = \tan^{-1}\left(\dfrac{\rho_u}{\sqrt{\rho_e^2+\rho_n^2}}\right) + n_{el} \\ \|\boldsymbol{\rho}\| = \sqrt{\rho_u^2+\rho_e^2+\rho_n^2} + n_\rho \end{cases} \tag{4.41}$$

where the azimuth (*az*), the elevation (*el*), and the range $\boldsymbol{\rho} = \begin{bmatrix}\rho_u & \rho_e & \rho_n\end{bmatrix}^T$ can be measured by the radar site on the ground with respect to the local observer coordinate system, ($\hat{u}-\hat{e}-\hat{n}$; "up, east and north"). The range can be related to the position vector in the inertial frame (***I-J-K***) by the coordinate transformation given by Eq. (4.42).

$$\begin{bmatrix}\rho_u \\ \rho_e \\ \rho_n\end{bmatrix} = \begin{bmatrix}\cos\lambda & 0 & \sin\lambda \\ 0 & 1 & 0 \\ -\sin\lambda & 0 & \cos\lambda\end{bmatrix}\begin{bmatrix}\cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x-\|\mathbf{R}\|\cos\lambda\cos\theta \\ y-\|\mathbf{R}\|\cos\lambda\sin\theta \\ z-\|\mathbf{R}\|\sin\lambda\end{bmatrix} \tag{4.42}$$

where $\|\mathbf{R}\| = 6378.1363\text{km}$ is the earth radius, λ and θ are the latitude and local sidereal time of the observer respectively, and n_{az}, n_{el}, and n_ρ are the white Gaussian measurement noise. The geometry of the observation model is shown in Figure 4.10.

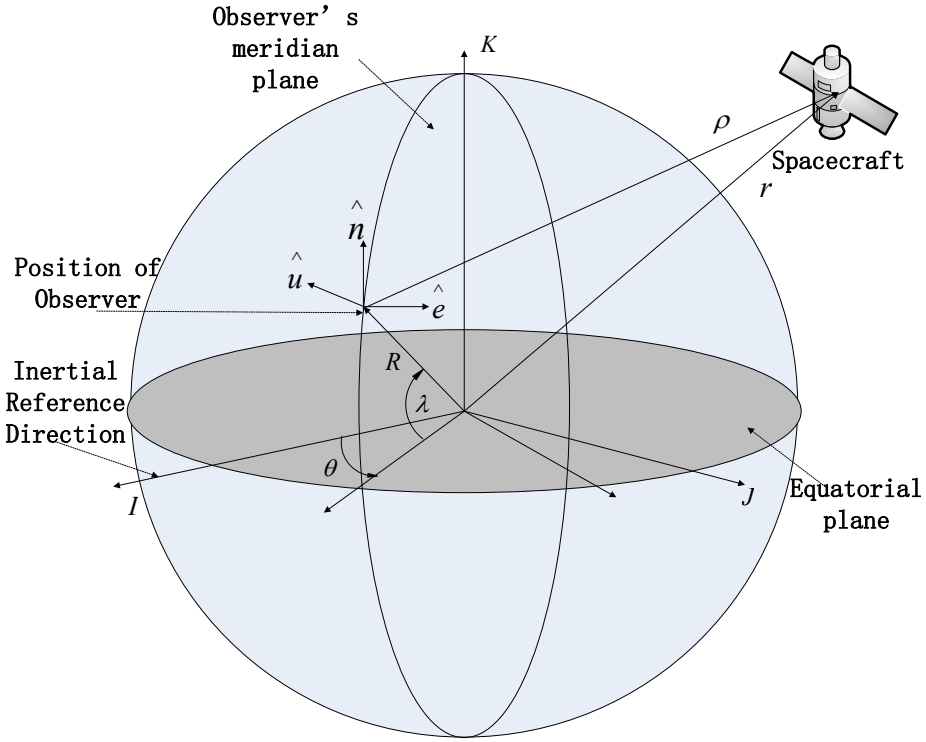

Figure 4.10 Illustration of the observation geometry

Generally, it is impossible to track a LEO satellite continuously from a single radar station. In the assumed simulation scenario, the reasonable track time is about 5 minutes. Hence, the simulation time is set to be 5 minutes and the measurement period is 5 seconds. The step size of 0.1 second and the fourth-order Runge-Kutta algorithm are used for dynamic propagation [52]. The latitude and the longitude of the radar site are -10.749^0 and -70.5983^0, respectively. The initial simulation time is March 29, 2011, 17:00:00. Moreover, the atmospheric drag parameters are $C_d = 2.2$, $A/m_s = 0.02\text{m}^2/\text{kg}$, and the atmospheric density ρ_d is calculated by the exponential model [51].

The state description of this orbit determination problem is $\boldsymbol{x}_0 = \left[\left(\boldsymbol{x}^p \right)^T \quad \left(\boldsymbol{x}^v \right)^T \right]^T = \left[x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z} \right]^T$. The initial true state value is assumed to be $\boldsymbol{x}_0 = \left[\left(\boldsymbol{x}_0^p \right)^T \quad \left(\boldsymbol{x}_0^v \right)^T \right]^T$. where $\left(\boldsymbol{x}_0^p \right)^T = \left[6949.599783, 1045.733299, 64.918535 \right]$km and $\left(\boldsymbol{x}_0^v \right)^T = \left[-0.902571, 5.697655, 4.841182 \right]$km/s are the initial true position and velocity of the satellite, respectively. The above orbital parameters, including the initial conditions, the radar site, and the track time are verified to be feasible using STK®, a well-accepted software tool for high-fidelity modeling of satellite orbits.

The level-2 and level-3 SGQFs are applied to this orbit determination problem and compared with the EKF, the UKF, the CKF, and the GHQF. The level-2 SGQ points and weights are computed using Eq. (2.46) and the level-3 SGQ points and weights are computed using Eqs. (3.24)-(3.26).

There are three cases of level-3 SGQFs as discussed in Section 3.2. Case 1 does not have any tunable parameters and uses the GHQ points $\hat{p}_1 = \hat{p}_2 = \hat{p}_3 = \sqrt{3}$; Case 2 has two tunable parameters, $\hat{p}_1$ (= $\hat{p}_2$) and $\hat{p}_3$; and Case 3 has three tunable parameters, $\hat{p}_1$, $\hat{p}_2$, and $\hat{p}_3$. The SGQF parameter $\hat{p}_1$ is used in both level-2 and level-3 SGQFs. The other two parameters are used only in the level-3 SGQFs. In other words, the level-2 SGFQ only has one tunable parameter $\hat{p}_1$ while the level-3 SGQFs have up to three tunable parameters.

To investigate the effect of the SGQ parameters on the filter performance, seven SGQFs with different parameters are tested. The GHQ points $\hat{p}_1 = \hat{p}_2 = \hat{p}_3 = \sqrt{3}$ are used as the reference points about which $\hat{p}_1$, $\hat{p}_2$, and $\hat{p}_3$ are varied and given in Table 4.3. Note that, for level-2 SGQFs, the SGQ points and weights are only determined by $\hat{p}_1$, and for level-3 SGQFs the locations and weights of most points are determined by $\hat{p}_1$

from Eqs. (3.24)-(3.26) . Therefore, the influence of $\hat{p}_1$ is more significant than the other two points.

In the following, the root-mean square errors (RMSEs) of SGQFs over 50 runs are compared with those of the EKF, UKF, CKF, and GHQF. The RMSEs of the position and velocity estimates at time *k* are defined as

$$\mathrm{RMSE}_{\mathrm{pos}}(k)=\sqrt{\frac{1}{N_{mc}}\sum_{n=1}^{N_{mc}}\left(\left(x_k-\hat{x}_k\right)^2+\left(y_k-\hat{y}_k\right)^2+\left(z_k-\hat{z}_k\right)^2\right)} \tag{4.43}$$

$$\mathrm{RMSE}_{\mathrm{vel}}(k)=\sqrt{\frac{1}{N_{mc}}\sum_{n=1}^{N_{mc}}\left(\left(\dot{x}_k-\dot{\hat{x}}_k\right)^2+\left(\dot{y}_k-\dot{\hat{y}}_k\right)^2+\left(\dot{z}_k-\dot{\hat{z}}_k\right)^2\right)} \tag{4.44}$$

where $N_{mc}=50$ is the number of simulation runs; $\begin{bmatrix}\hat{x}_k & \hat{y}_k & \hat{z}_k\end{bmatrix}^T$ and $\begin{bmatrix}\dot{\hat{x}}_k & \dot{\hat{y}}_k & \dot{\hat{z}}_k\end{bmatrix}^T$ are the estimated position and velocity at time *k,* respectively.

Table 4.3 Parameters of SGQFs

	$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$
1st SGQF (Case 1)	$\sqrt{3}$	$\sqrt{3}$	$\sqrt{3}$
2nd SGQF (Case 2)	1.71	1.71	2.50
3rd SGQF (Case 2)	1	1	2.5
4th SGQF (Case 2)	0.5	0.5	2
5th SGQF (Case 3)	1.71	1.00	2.50
6th SGQF (Case 3)	1	2.5	1.5
7th SGQF (Case 3)	0.5	2	1

Two different scenarios, with small initial estimation errors and large initial estimation errors, are used to test the filter performance.

1.1.1 The First Scenario: Small Initial Estimation Errors

In this scenario, the initial estimates are assumed with small errors $\left(\hat{\boldsymbol{x}}_0^p\right)^T = \left[6989.157085, 1061.379082, 117.351971\right]\text{km}$ and $\left(\hat{\boldsymbol{x}}_0^v\right)^T = \left[-0.899065, 5.291087, 5.005861\right]\text{km/s}$. The initial covariance is $\mathbf{P}_0 = \text{diag}\left(\left[\left[10^2, 10^2, 10^2\right](\text{km})^2, \left[10^{-2}, 10^{-2}, 10^{-2}\right](\text{km/s})^2\right]\right)$. The reasonable process and measurement noise covariances are assumed to be [52].

$\mathbf{Q}_k = \text{diag}\left(\left[0, 0, 0, 10^{-17}(\text{km/s})^2, 10^{-17}(\text{km/s})^2, 10^{-17}(\text{km/s})^2\right]\right)$ and

$\mathbf{R}_k = \text{diag}\left(\left[\left(0.015^\circ\right)^2, \left(0.015^\circ\right)^2, 0.025^2(\text{km})^2\right]\right)$ respectively.

The SGQFs (level-2 and level-3) using the parameters in Table 4.3 are tested and compared with the EKF, UKF/CKF, and GHQF. Since κ is a tunable parameter for UKF, a range of $\kappa = -3, -2, -1, 0, 1, 2, 3$ are used for the test. The position and velocity RMSEs are shown in Figure 4.11 and Figure 4.12 respectively. Because there is no discernible difference among all SGQFs (level-2 or level-3) and UKFs are identical to level-2 SGQFs (*Theorem 3.3*), only one of the level-2 SGQFs and one of the level-3 SGQFs are given in Figure 4.11 and Figure 4.12. The simulation results show that SGQFs are slightly more accurate than EKF. The level-2 SGQFs, level-3 SGQFs, and GHQF all have such close performance that their RMSEs are indistinguishable. Therefore, the point-based methods (UKFs, SGQFs, and GHQF) do not show significant advantages over EKF when the initial estimation errors are small.

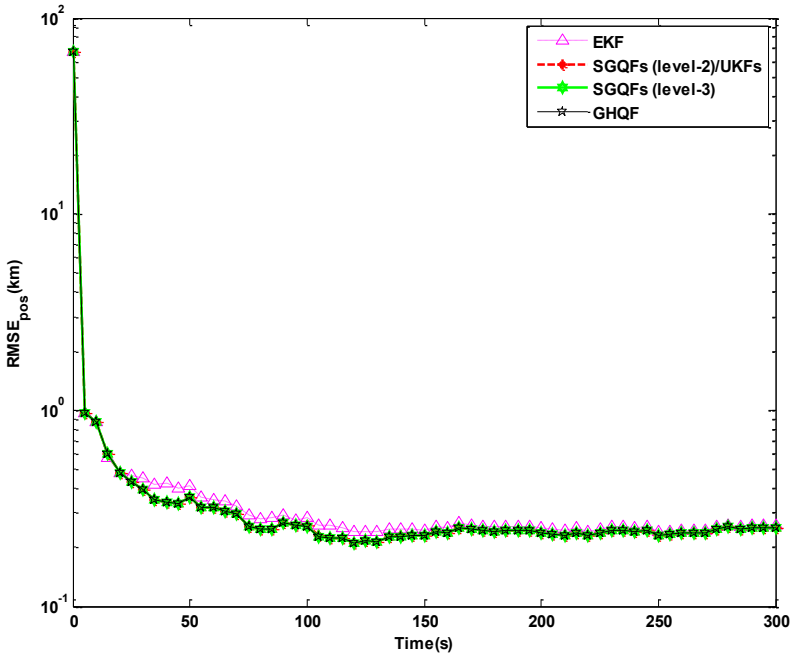

Figure 4.11 RMSEs of the position with small initial estimation errors

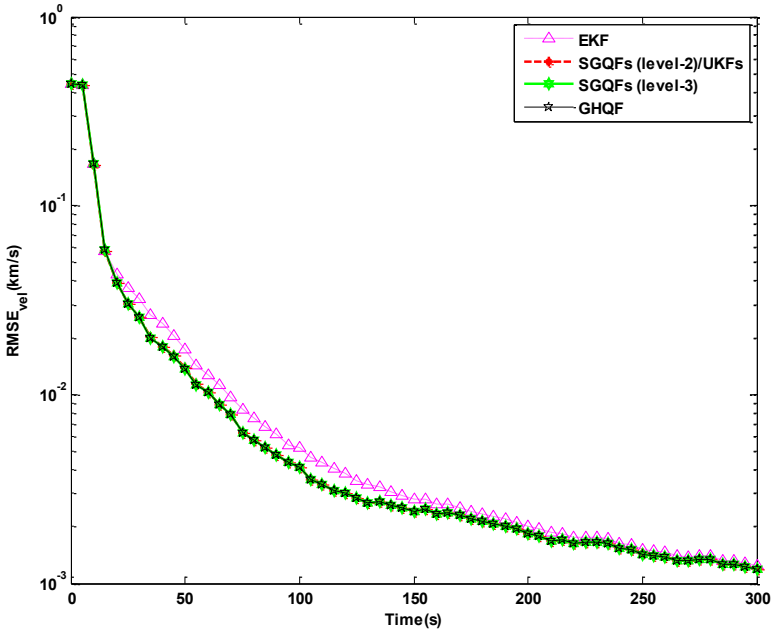

Figure 4.12 RMSEs of the velocity with small initial estimation errors

4.2.2 The Second Scenario: Large Initial Estimation Errors

In this scenario, the initial estimates are assumed with large errors $\left(\hat{\boldsymbol{x}}_0^p\right)^T = [7252.009273, 1358.407862, 383.904071]$km and $\left(\hat{\boldsymbol{x}}_0^v\right)^T = [-0.613101, 5.991868, 5.138553]$km . The initial covariance is $\mathbf{P}_0 = \text{diag}\left(\left[\left[10^4, 10^4, 10^4\right](\text{km})^2, \left[10^{-2}, 10^{-2}, 10^{-2}\right](\text{km/s})^2\right]\right)$.

4.2.2.1 Results of UKFs and level-2 SGQFs

In this section, we compare the SGQF at accuracy level-2 with EKF and GHQF. The first group of tested level-2 SGQFs is the UKFs with a range of the tunable parameters $\kappa = -3, -2, -1, 0, 1, 2, 3$. Recall that the UKF is identical to the level-2 SGQF with $\hat{p}_1 = \sqrt{n+\kappa}$ (*Theorem 3.3* and Eq. (3.30)). The role of the tunable parameter κ for UKF is the same as the tunable parameter $\hat{p}_1$ for SGQF. The RMSEs of the position and velocity estimates for EKF, UKFs, and GHQF are shown in Figure 4.13 and Figure 4.14, respectively.

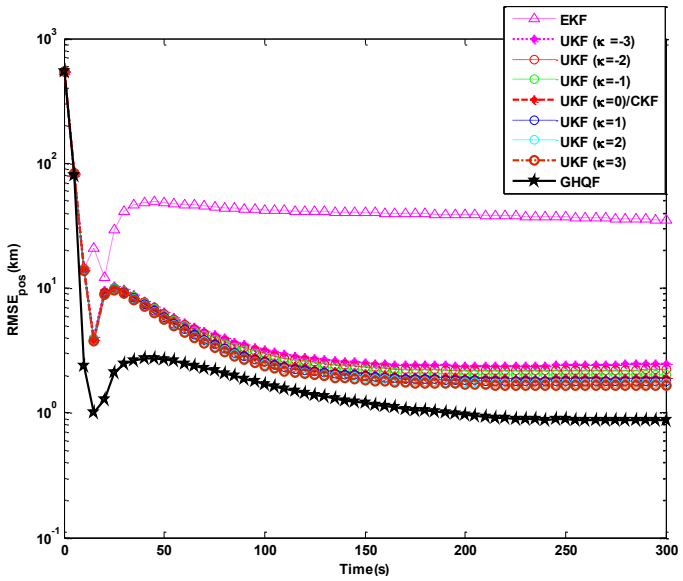

Figure 4.13 RMSEs of the position with large initial estimation errors: EKF, UKFs, and GHQF

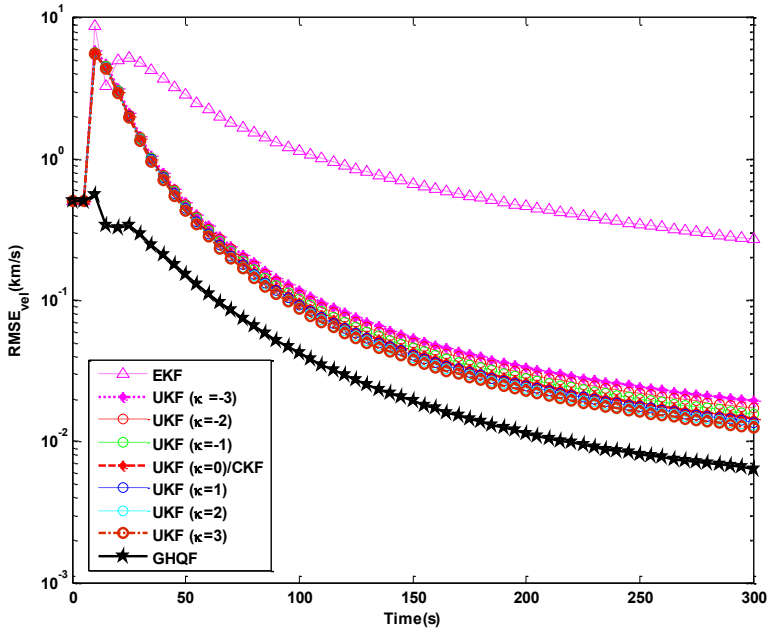

Figure 4.14 RMSEs of the velocity with large initial estimation errors: EKF, UKFs, and GHQF

As can be seen, UKFs and GHQF are much more accurate than EKF for large initial estimation errors. Additionally, all the UKFs for different κ parameters exhibit close performance. Note that the suggested parameter of κ is $\kappa = 3 - n = -3$ [35], and the CKF is identical to the UKF when $\kappa = 0$. For this particular application, UKF has the best performance when $\kappa = 3$. The CKF is slightly better than the UKF with the suggested value of $\kappa = -3$. GHQF is obviously more accurate than all UKFs.

In the above group of tested UKFs or level-2 SGQFs, the parameter $\hat{p}_1$ corresponding to the same as κ used for UKF are all greater than or equal to $\sqrt{3}$. So in the second group of tested level-2 SGQFs, we use the values of $\hat{p}_1$ that are smaller than $\sqrt{3}$ for further comparison. These parameters of $\hat{p}_1$ correspond to the 1st SGQF through 4th SGQFs as shown in Table 4.3. Since all the UKFs and CKF do not show much difference in performance, we will select the CKF and the UKF with the suggested

parameter $\kappa = -3$ as the references for comparison with other level-2 SGQFs, i.e. the 1st SGQF (level-2) to the 4th SGQF (level-2) in Table 4.3.

The RMSEs of the position and velocity for EKF, level-2 SGQFs, and GHQF are shown in Figure 4.15 and Figure 4.16. As can be seen, all level-2 SGQFs are much more accurate than the EKF but not as accurate as GHQF. Note that the 1st SGQF (level-2) is the same as the UKF with the suggested parameter $\kappa = -3$.

In summary, all the level-2 SGQFs including UKF and CKF with different parameter values demonstrate close performance, which is more accurate than EKF but not as accurate as GHQF.

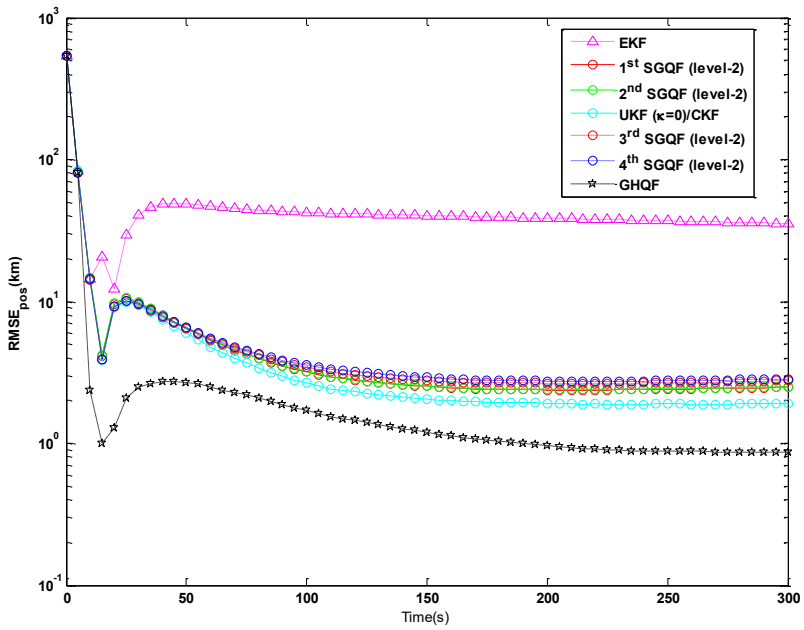

Figure 4.15 RMSEs of the position with large initial estimation errors: EKF, level-2 SGQFs, and GHQF

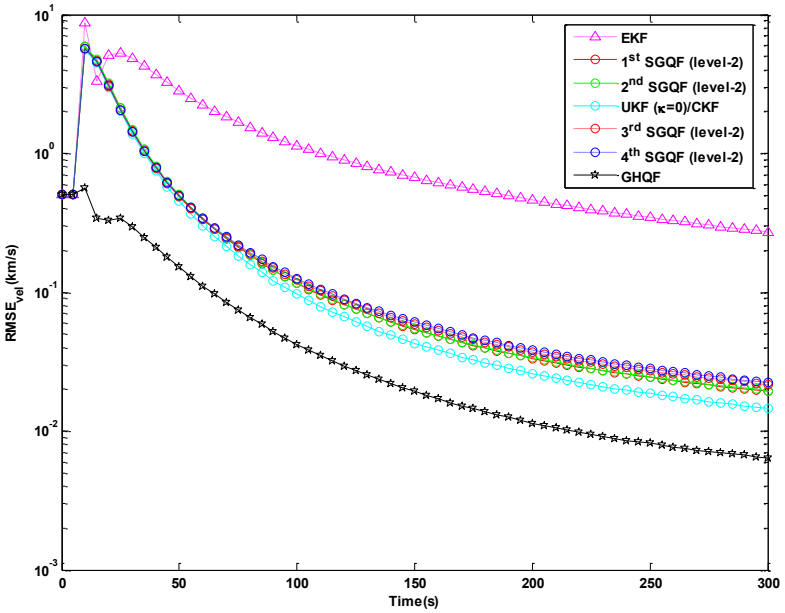

Figure 4.16 RMSEs of the velocity with large initial estimation errors: EKF, level-2 SGQFs, and GHQF

4.2.2.2 Results of level-3 SGQFs

In this section, the level-3 SGQFs with different parameters of $\hat{p}_1, \hat{p}_2$, and $\hat{p}_3$ corresponding to the three cases in Table 4.3, is compared with EKF, UKF ($\kappa = 3$ with the best performance), and GHQF. The RMSEs of the position and velocity are shown in Figure 4.17 and Figure 4.18.

It can be seen that all level-3 SGQFs are more accurate than the EKF and level-2 SGQF/UKF. All level-3 SGQFs are overlapped and have no discernible difference, which implies that these level-3 SGQFs are not sensitive to the SGQ parameters. Note that the level-2 SGQ can approximate the integral calculations up to the 3rd order polynomials whereas level-3 SGQ can be exact up to the 5th order polynomials (*Theorem 3.1*). From the simulation tests, it indicates that the 3rd order polynomial approximation is not accurate enough, which is the reason that one can see the performance difference (Section 4.2.1) using different parameters of $\hat{p}_1$ (or κ for UKF). However, the level-3 SGQ that is

exact up to the 5th order polynomials is accurate enough to capture the nonlinear characteristics and initial estimate uncertainty of this orbit estimation problem. Note that no matter how the parameters $\hat{p}_1$, $\hat{p}_2$, $\hat{p}_3$ are tuned, the 5th order accuracy can be guaranteed and thus the performance will not exhibit noticeable difference.

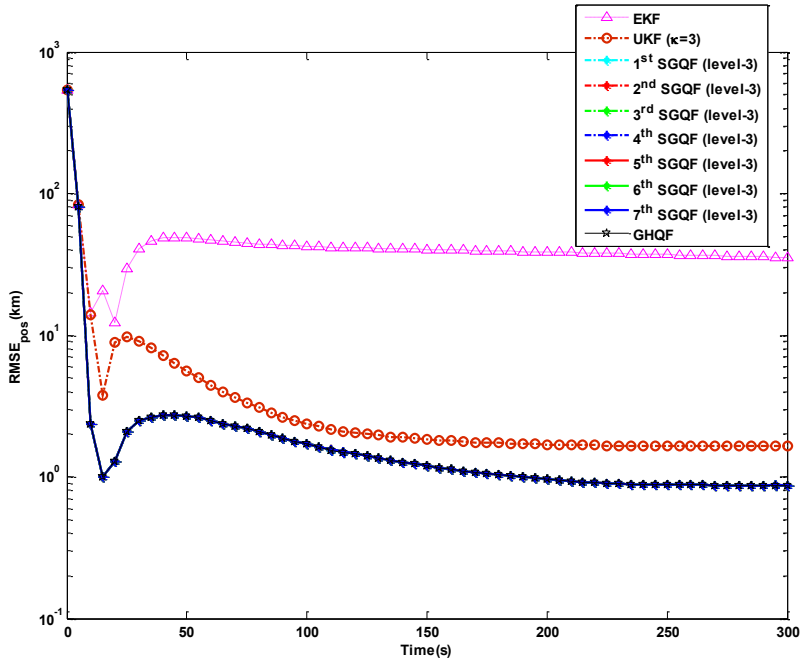

Figure 4.17 RMSEs of the position with large initial estimation errors: EKF, UKF, level-3 SGQFs, and GHQF

Figure 4.17 and Figure 4.18 also show that the level-3 SGQFs have the same performance as the GHQF (m_L=3). However, the level-3 SGQFs uses 73, 85, and 97 points for the Cases 1, 2, and 3, respectively whereas the GHQF uses 3^6 (=729) points. No comparison was made using the GHQF with higher accuracy because it would require a large number of GHQ points, for example, 5^6 (=15625) for m_L=5. The number of points for different SGQFs and GHQF are shown in Table 4.4, which indicates that the SGQF is computationally much more efficient than the GHQF.

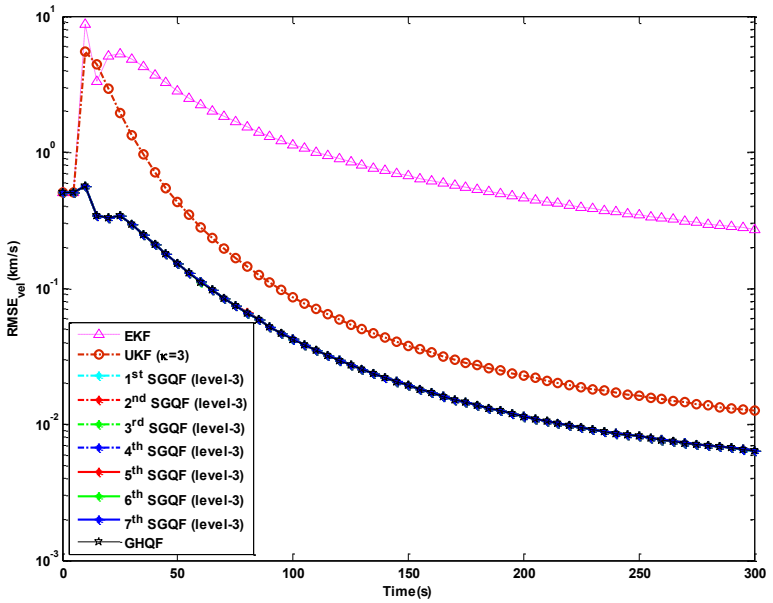

Figure 4.18 RMSEs of the velocity with large initial estimation errors: EKF, UKF, level-3 SGQFs, and GHQF

Table 4.4 Number of Points for different SGQFs and GHQF

	SGQF (Case 1)	SGQF (Case 2)	SGQF (Case 3)	GHQF $m_L = 3$
level-2	13	13	13	729
level-3	73	85	97	

From the above simulation results, Figure 4.13-4.18, it can be seen that the level-3 SGQFs demonstrate the best performance and are good enough for this orbit determination problem. It is worth noting that for other applications, higher level-SGQF may be desirable.

In this section, GHQ points are used as the reference points to tune the parameters of SGQ. If the three-point univariate GHQ is used for I_2 and I_3, the reference points are

$\hat{p}_1 = \hat{p}_2 = \hat{p}_3 = \sqrt{3}$. Higher accuracy GHQ points can be used for I_2 and I_3 as well. For example, 3 and 5 univariate GHQ points can be used for I_2 and I_3, respectively, with the reference points being $\hat{p}_1 = \sqrt{3}$, $\hat{p}_2 = 1.3556$ and $\hat{p}_3 = 2.8570$. The parameters of SGQ can be tuned around these reference points. For SGQ of higher than three accuracy levels, which need more points to match higher moments, more than 3 univariate GHQ points should be used as the reference points. In addition, the GHQ points are optimal for the integrals when the integrand function is of the special form that is the multiplication of a polynomial function and the Gaussian pdf. For a general form that is the multiplication of a general nonlinear function and a non-Gaussian pdf, the GHQ points may not be optimal and the tunable parameter $\hat{p}_i$ can play an important role in the performance of the SGQF.

4.3 Spacecraft Relative Navigation

The relative navigation of spacecraft is important to many space missions, such as rendezvous, docking, and formation flying [93, 94]. A variety of technologies have been proposed to determine the relative attitude and orbit between spacecraft. GPS was used in [95, 96] for relative navigation. However, GPS signals are difficult to obtain in many situations like deep-space missions. To circumvent the GPS limitations, many self-dependent measurement systems have been proposed. The inner-formation gravity measurement satellite system was used in [97] and laser radar based relative navigation was proposed in [98]. The vision-based navigation system has recently drawn much attention [93, 99]. There are many advantages of the vision-based navigation system, such as small sensor size and wide sensor field of view. More features can be found in [100].

In this section, the relative navigation of two spacecraft is considered. The relative navigation problem includes the estimation of the relative attitude and the relative orbit. In addition, the gyro bias of the chief and the deputy can also be obtained or corrected. The dimension of the system state is 16 and is thus a high dimension estimation problem. In the vision-based navigation system, beacons (specific light sources) are used to achieve a selective vision and position-sensing diode based sensors placed in the focal plane are used to locate the beacons (light sources) [100].

The classical extended Kalman filter (EKF) has been used in [54, 93] to estimate the attitude and orbit simultaneously. However, the EKF may not be accurate when the initial uncertainty is large. Many point-based Gaussian approximation filters proposed in recent years can be used to improve the estimation accuracy. Among these filters, the unscented Kalman filter (UKF) is probably the most widely used [35, 36]. The UKF has been employed to solve the relative navigation problem in [101]. Although the UKF is more accurate than the EKF, it is less accurate than the Gauss-Hermite quadrature filter (GHQF) [56]. Nevertheless, the GHQF cannot be used to for the relative navigation problem because the number of points required by the GHQF increases exponentially with the increase of the dimension, which is the curse-of-dimensionality problem. The sparse Gauss-Hermite quadrature filter (SGHQF) [25] can be used to alleviate this problem while maintaining close performance to the GHQF. In addition, we have shown that the SGHQF can achieve higher accuracy than the UKF by using additional quadrature points [25]. In this section, the SGHQF is used to solve the vision-based relative navigation problem.

4.3.1 Relative Navigation Model

In this section, the relative attitude kinematics and the relative orbit dynamics of two spacecraft, as well as the vision-based measurement model, are briefly reviewed.

4.3.1.1 Relative Attitude Kinematics

The relative attitude is described by the relative quaternion and its kinematics at time $k-1$, and is given by [93]

$$\mathbf{q}_k = \Omega\left(\boldsymbol{\omega}_{d,k-1}\right)\Gamma\left(\boldsymbol{\omega}_{c,k-1}\right)\mathbf{q}_{k-1} \tag{4.45}$$

with

$$\Omega\left(\boldsymbol{\omega}_{d,k-1}\right) = \begin{bmatrix} \cos\left(\frac{1}{2}\left\|\boldsymbol{\omega}_{d,k-1}\right\|\Delta t\right) I_{3\times 3} - \left[\boldsymbol{\psi}_{k-1}\times\right] & \boldsymbol{\psi}_{k-1} \\ -\boldsymbol{\psi}_{k-1}^T & \cos\left(\frac{1}{2}\left\|\boldsymbol{\omega}_{d,k-1}\right\|\Delta t\right) \end{bmatrix} \tag{4.46}$$

$$\Gamma\left(\boldsymbol{\omega}_{c,k-1}\right) = \begin{bmatrix} \cos\left(\frac{1}{2}\left\|\boldsymbol{\omega}_{c,k-1}\right\|\Delta t\right) I_{3\times 3} - \left[\boldsymbol{\delta}_{k-1}\times\right] & \boldsymbol{\delta}_{k-1} \\ -\boldsymbol{\delta}_{k-1}^T & \cos\left(\frac{1}{2}\left\|\boldsymbol{\omega}_{c,k-1}\right\|\Delta t\right) \end{bmatrix} \tag{4.47}$$

and

$$\boldsymbol{\psi}_{k-1} = \frac{\sin\left(\frac{1}{2}\left\|\boldsymbol{\omega}_{d,k-1}\right\|\Delta t\right)\boldsymbol{\omega}_{d,k-1}}{\left\|\boldsymbol{\omega}_{d,k-1}\right\|} \tag{4.48}$$

$$\boldsymbol{\delta}_{k-1} = \frac{\sin\left(\frac{1}{2}\left\|\boldsymbol{\omega}_{c,k-1}\right\|\Delta t\right)\boldsymbol{\omega}_{c,k-1}}{\left\|\boldsymbol{\omega}_{c,k-1}\right\|} \tag{4.49}$$

where $\boldsymbol{\omega}_{c,k-1}$ and $\boldsymbol{\omega}_{d,k-1}$ are angular velocities of the chief and the deputy at the time $k-1$, respectively. Δt is the sample interval and $\left[\boldsymbol{\psi}_{k-1}\times\right]$ is a cross product matrix. Assume that $\boldsymbol{\psi}_{k-1} = \left[\psi_{1,k-1}, \psi_{2,k-1}, \psi_{3,k-1}\right]^T$ and $\left[\boldsymbol{\psi}_{k-1}\times\right]$ is defined by

$$\left[\boldsymbol{\psi}_{k-1}\times\right] = \begin{bmatrix} 0 & -\psi_{3,k-1} & \psi_{2,k-1} \\ \psi_{3,k-1} & 0 & -\psi_{1,k-1} \\ -\psi_{2,k-1} & \psi_{1,k-1} & 0 \end{bmatrix} \tag{4.50}$$

The model used to measure the angular velocity is given by [50]

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{\beta} + \boldsymbol{\varepsilon}_v \tag{4.51}$$

$$\dot{\boldsymbol{\beta}} = \boldsymbol{\varepsilon}_u \tag{4.52}$$

where $\tilde{\boldsymbol{\omega}}$ and $\boldsymbol{\omega}$ are the continuous-time measured and true angular velocity, respectively. $\boldsymbol{\beta}$ is the gyro bias, and $\boldsymbol{\varepsilon}_v$ and $\boldsymbol{\varepsilon}_u$ are independent white Gaussian noise with zero mean and covariance $\sigma_v^2 I_{3\times3}$ and $\sigma_u^2 I_{3\times3}$, respectively.

In the standard filtering formulation, given the post-update gyro drift rate $\hat{\boldsymbol{\beta}}_{k-1}^{+}$, the estimated angular velocity $\hat{\boldsymbol{\omega}}_{k-1}$ is given by [50, 93]

$$\hat{\boldsymbol{\omega}}_{k-1}^{+} = \tilde{\boldsymbol{\omega}}_{k-1} - \hat{\boldsymbol{\beta}}_{k-1}^{+} \tag{4.53}$$

where $\tilde{\boldsymbol{\omega}}_{k-1}$ is the measured angular velocity at time $k-1$.

The prediction of the gyro drift is given by

$$\hat{\boldsymbol{\beta}}_{k}^{-} = \hat{\boldsymbol{\beta}}_{k-1}^{+} \tag{4.54}$$

4.3.1.2 Relative Orbit Dynamic Equations

In this section, we consider a circular or near circular orbit. In this case, Clohessy-Wiltshire equations can be obtained [51]

$$\ddot{x} - 2\omega_r \dot{y} - 3\omega_r^2 x = 0 \tag{4.55}$$

$$\ddot{y} + 2\omega_r \dot{x} = 0 \tag{4.56}$$

$$\ddot{z} + \omega_r^2 z = 0 \tag{4.57}$$

where $[x, y, z]^T$ and $[\dot{x}, \dot{y}, \dot{z}]^T$ are the relative position and velocity vector, respectively and ω_r is the mean orbital rate, given by

$$\omega_r = \sqrt{\frac{\mu}{a^3}} \tag{4.58}$$

with μ and a being the standard gravitational parameter and the radius of the chief, respectively. The radius of the chief is assumed to be known.

Define the state variables of the relative orbit as $\mathbf{x}_r = \left[x, y, z, \dot{x}, \dot{y}, \dot{z}\right]^T$.

Considering the perturbation noise, the state equation becomes

$$\dot{\mathbf{x}}_r = \boldsymbol{f}\left(\mathbf{x}_r\right) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ 2\omega_r \dot{y} + 3\omega_r^2 x + v_x \\ -2\omega_r \dot{x} + v_y \\ -\omega_r^2 z + v_z \end{bmatrix} \tag{4.59}$$

where v_x, v_y, and v_z are the Gaussian noise with variances of σ_x^2, σ_y^2, and σ_z^2, respectively.

4.3.1.3 Vision-Based Measurement Model

The vision-based measurement can be modeled as [93]

$$\tilde{\mathbf{b}}_i = A(\mathbf{q})\mathbf{r}_i + \mathbf{n}_i \qquad i = 1, 2, \cdots, N_s \tag{4.60}$$

with $A(\mathbf{q}) = \Theta^T(\mathbf{q})\psi(\mathbf{q})$;

$$\Theta(\mathbf{q}) = \begin{bmatrix} q_4 I_{3\times 3} + [\boldsymbol{\rho}\times] \\ -\boldsymbol{\rho}^T \end{bmatrix} \tag{4.61}$$

$$\psi(\mathbf{q}) = \begin{bmatrix} q_4 I_{3\times 3} - [\boldsymbol{\rho}\times] \\ -\boldsymbol{\rho}^T \end{bmatrix} \tag{4.62}$$

where $\mathbf{q} = \left[\boldsymbol{\rho}^T, q_4\right]^T$ is the relative quaternion in which $\boldsymbol{\rho}$ is the vector part and q_4 is the scalar part; N_s is the number of sensors; $\tilde{\mathbf{b}}_i$ denotes the measured value by the i^{th} sensor and $\mathbf{r}_i$ is given by [93]

$$\mathbf{r}_i = \frac{1}{\sqrt{\left(X_i - x\right)^2 + \left(Y_i - y\right)^2 + \left(Z_i - z\right)^2}} \begin{bmatrix} X_i - x \\ Y_i - y \\ Z_i - z \end{bmatrix} \tag{4.63}$$

Note that $\left(X_i, Y_i, Z_i\right)$ is the known object space location of the i^{th} beacon and $\mathbf{n}_i$ is white Gaussian noise.

4.3.2 Relative Attitude And Orbit Estimation

The state vector of the relative navigation, including the relative attitude and the relative orbit is given by $\mathbf{x} = \left[\mathbf{x}_a^T, \mathbf{x}_r^T\right]^T$ where $\mathbf{x}_a \in \mathbb{R}^9$ and $\mathbf{x}_r \in \mathbb{R}^6$ denote the relative attitude vector and the relative orbit vector, respectively. Many parameters can be used to represent the three-axis attitude as discussed in Section 4.1.

In this chapter, unconstrained MRPs are used to represent attitude errors in the filtering algorithm. Given the attitude error represented by MPRs $\boldsymbol{\delta\rho}$, the error quaternion $\boldsymbol{\delta q} = \left[\boldsymbol{\delta\rho}^T, \delta q_4\right]^T$ is given by

$$\delta q_4 = \frac{-a_m \left\|\boldsymbol{\delta\rho}\right\|^2 + f_c\sqrt{f_c^2 + (1-a_m)\left\|\boldsymbol{\delta\rho}\right\|^2}}{f_c^2 + \left\|\boldsymbol{\delta\rho}\right\|^2} \tag{4.64}$$

$$\boldsymbol{\delta\rho} = f_c^{-1}\left(a_m + \delta q_4\right)\boldsymbol{\delta\rho} \tag{4.65}$$

where a_m and f_c are two parameters.

Given the error quaternion $\boldsymbol{\delta q}$, the MRPs $\boldsymbol{\delta\rho}$ is given by

$$\boldsymbol{\delta\rho} = f_c \frac{\delta\tilde{\mathbf{q}}}{a_m + \delta q_4} \tag{4.66}$$

In this section, we use $a_m = 1$ and $f_c = 4$. Define the state of the relative attitude at time k as $\hat{\mathbf{x}}_{a,k} = \left[\boldsymbol{\delta\rho}_k^T \quad \boldsymbol{\beta}_{c,k}^T \quad \boldsymbol{\beta}_{d,k}^T\right]^T$, where $\boldsymbol{\delta\rho}_k$, $\boldsymbol{\beta}_{c,k}$, and $\boldsymbol{\beta}_{d,k}$ are the attitude error, the gyro bias of the chief, and the gyro bias of the deputy, respectively.

4.3.2.1 Relative Navigation Algorithm

Given the initial estimate $\hat{\mathbf{x}}_0 = \left[\hat{\mathbf{x}}_{a,0}^T, \hat{\mathbf{x}}_{r,0}^T\right]^T$, $\hat{\mathbf{q}}_0$ and the initial covariance $\mathbf{P}_0$, the point-based Gaussian approximation relative navigation filtering can be summarized as follows.

Prediction

1) The transformed points $\boldsymbol{\xi}_{k-1|k-1}(i) = \left[\left(\boldsymbol{\xi}_{a,k-1|k-1}(i)\right)^T, \left(\boldsymbol{\xi}_{r,k-1|k-1}(i)\right)^T\right]^T$ are calculated by Eq. (2.31). Note that

$\boldsymbol{\xi}_{a,k-1|k-1}(i)=\left[\boldsymbol{\xi}^{\boldsymbol{\delta p}}_{a,k-1|k-1}(i)^T \quad \boldsymbol{\xi}^{\hat{\boldsymbol{\beta}}_c}_{a,k-1|k-1}(i)^T \quad \boldsymbol{\xi}^{\hat{\boldsymbol{\beta}}_d}_{a,k-1|k-1}(i)^T\right]^T$ with i being the point index and $\boldsymbol{\xi}^{\boldsymbol{\delta p}}_{a,k-1|k-1}(1)=\mathbf{0}_{3\times1}$ [50]; $\left(\xi_{r,k-1|k-1}(i)\right)^T$ are the transformed SGHQ points corresponding to the six dimensional state $\mathbf{x}_r$. $\boldsymbol{\xi}^{\boldsymbol{\delta p}}_{a,k-1|k-1}(i)$ are then transformed into error quaternion $\boldsymbol{\delta q}^{pre}_{k-1|k-1}(i)$ by Eqs. (4.64) and (4.65). The superscript „*pre*" denotes the prediction step.

2) The transformed quaternions $\hat{\mathbf{q}}^{pre}_{k-1|k-1}(i)$ can be obtained by $\hat{\mathbf{q}}^{pre}_{k-1|k-1}(i)=\boldsymbol{\delta q}^{pre}_{k-1|k-1}(i)\odot\hat{\mathbf{q}}_{k-1|k-1}$, where $\odot$ denotes the quaternion product.

3) The predicted quaternions $\hat{\mathbf{q}}^{pre}_{k|k-1}(i)$ are predicted by Eq. (4.45). Note, $\hat{\boldsymbol{\omega}}_{c,k-1}$ and $\hat{\boldsymbol{\omega}}_{d,k-1}$ are obtained by Eq. (4.53).

4) Error quaternions $\boldsymbol{\delta}\hat{\mathbf{q}}^{pre}_{k|k-1}(i)$ are calculated by $\boldsymbol{\delta}\hat{\mathbf{q}}^{pre}_{k|k-1}(i)=\hat{\mathbf{q}}^{pre}_{k|k-1}(i)\odot\left[\hat{\mathbf{q}}^{pre}_{k|k-1}(1)\right]^{-1}$. Then, the predicted points $\boldsymbol{\xi}^{\boldsymbol{\delta p}}_{a,k|k-1}(i)$ can be calculated by Eq. (4.66). The predicted points $\boldsymbol{\xi}^{\hat{\boldsymbol{\beta}}_c}_{a,k|k-1}(i)$ and $\boldsymbol{\xi}^{\hat{\boldsymbol{\beta}}_d}_{a,k|k-1}(i)$ are given by Eq. (4.54) and $\boldsymbol{\xi}_{r,k|k-1}(i)$ is obtained by the relative orbit dynamics in Eqs. (4.55)-(4.57). Hence, $\boldsymbol{\xi}_{k|k-1}(i)=\left[\left(\xi_{a,k|k-1}(i)\right)^T,\left(\xi_{r,k|k-1}(i)\right)^T\right]^T$ can be obtained.

5) The mean and covariance are calculated by Eqs. (2.29)-(2.30). Then the first three mean values are transformed into the error quaternion $\boldsymbol{\delta q}_{k|k-1}$. The predicted quaternion $\hat{\mathbf{q}}_{k|k-1}$ is obtained by $\hat{\mathbf{q}}_{k|k-1}=\delta\mathbf{q}_{k|k-1}\odot\hat{\mathbf{q}}^{pre}_{k|k-1}(1)$

Update

1) Similarly, the transformed points $\tilde{\boldsymbol{\xi}}_{k|k-1}(i)=\left[\left(\tilde{\boldsymbol{\xi}}_{a,k|k-1}(i)\right)^T,\left(\tilde{\boldsymbol{\xi}}_{r,k|k-1}(i)\right)^T\right]^T$ are generated by Eq. (2.35). Then $\tilde{\boldsymbol{\xi}}^{\boldsymbol{\delta p}}_{a,k|k-1}(i)$ can be transformed into the error quaternion $\boldsymbol{\delta q}^{upd}_{k|k-1}(i)$ by Eqs. (4.64) and (4.65). The superscript „*upd*" denotes the update step; $\tilde{\boldsymbol{\xi}}^{\boldsymbol{\delta p}}_{a,k|k-1}(1)=\mathbf{0}_{3\times1}$ [50].

2) Calculate quaternions $\hat{\mathbf{q}}^{upd}_{k|k-1}(i)=\boldsymbol{\delta q}^{upd}_{k|k-1}(i)\odot\hat{\mathbf{q}}_{k|k-1}$.

3) Given the measurement values, the state $\hat{\mathbf{x}}_{k|k}$ and covariance $\mathbf{P}_{k|k}$ can be updated by Eqs. (2.21)-(2.23) with $\hat{\mathbf{q}}_{k|k-1}^{upd}(i)$, $\tilde{\boldsymbol{\xi}}_{k|k-1}(i)$, $\tilde{\mathbf{b}}_i$ and $\mathbf{R}_k$.

4) Calculate the error quaternion $\boldsymbol{\delta}\mathbf{q}_{k|k}$ by Eqs. (4.64) and (4.65). and the updated quaternion $\hat{\mathbf{q}}_{k|k} = \boldsymbol{\delta}\mathbf{q}_{k|k} \odot \hat{\mathbf{q}}_{k|k-1}$.

4.3.2.2 Approximation of The Process Noise

The covariance of discrete-time noise process $\mathbf{Q}_{k-1}$ used in filtering is intractable because it depends on the attitude matrix [93]. A numerically approximated discrete-time process noise can be obtained as follows.

The error-state dynamics of the relative navigation problem is given by [93]

$$\Delta\dot{\mathbf{x}} = \mathbf{F}\Delta\mathbf{x} + \mathbf{G}\mathbf{v}_c \tag{4.67}$$

where

$$\mathbf{F} = \begin{bmatrix} -[\hat{\boldsymbol{\omega}}_d \times] & A(\hat{\mathbf{q}}) & -I_{3\times3} & 0_{3\times6} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times6} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times6} \\ 0_{6\times3} & 0_{6\times3} & 0_{6\times3} & \left.\dfrac{\partial \boldsymbol{f}(\mathbf{x}_r)}{\partial \mathbf{x}_r}\right|_{\mathbf{x}_r} \end{bmatrix} \tag{4.68}$$

$$\mathbf{G} = \begin{bmatrix} A(\hat{\mathbf{q}}) & -I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \end{bmatrix} \tag{4.69}$$

The covariance of the process noise $\mathbf{v}_c$ is given by

$$\mathbf{Q}_c = \mathrm{diag}\left(\left[\left[\sigma_{cv}^2 I_{3\times3}\right],\left[\sigma_{dv}^2 I_{3\times3}\right],\left[\sigma_{cu}^2 I_{3\times3}\right],\left[\sigma_{du}^2 I_{3\times3}\right],\sigma_x^2,\sigma_y^2,\sigma_z^2\right]\right) \tag{4.70}$$

where $\sigma_{cv}^2 I_{3\times3}$ and $\sigma_{cu}^2 I_{3\times3}$ are covariance matrices corresponding to gyro noises in Eqs. (4.51) and (4.52) of the chief, respectively. Similarly, $\sigma_{dv}^2 I_{3\times3}$ and $\sigma_{du}^2 I_{3\times3}$ are covariance

matrices corresponding to gyro noises of the deputy. σ_x^2, σ_y^2, and σ_z^2 are covariances of Gaussian process noises in the relative orbit dynamics (Eq. (4.59)).

Then, the approximated discrete-time $\mathbf{Q}$ is given by [93]

$$\mathbf{Q} = \mathbf{B}_{22}^T \mathbf{B}_{12} \tag{4.71}$$

where $\mathbf{B}_{12}$ and $\mathbf{B}_{22}$ can be obtained from the following equations.

$$e^{\mathbf{A}} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} \tag{4.72}$$

with

$$\mathbf{A} = \begin{bmatrix} -\mathbf{F} & \mathbf{G}\mathbf{Q}_c\mathbf{G}^T \\ \mathbf{0} & \mathbf{F}^T \end{bmatrix} \Delta t \tag{4.73}$$

Note that $e^{(\cdot)}$ is the matrix exponential operator and that the discrete-time process noise at time $k-1$ can be obtained using Eq. (4.71).

4.3.3 Simulation Results and Analysis

In this section, simulation results are presented to compare the SGHQF with the EKF, the UKF, and the CKF. In the simulation scenario, the chief orbit radius is assumed to be 7,278,136 m. The initial estimated states are generated randomly by adding random errors with the normal distribution $N(\mathbf{0}, \mathbf{P}_0)$ to the initial true value

$\mathbf{x}_0 = \left[\mathbf{q}_0^T, \boldsymbol{\beta}_{c,0}^T, \boldsymbol{\beta}_{d,0}^T, \mathbf{x}_{r,0}^T\right]^T$, where

$\mathbf{q}_0 = \left[-0.011108, 0.707019, 0.58552, 0.396443029419108\right]^T$,

$\mathbf{x}_{r,0} = \left[200\text{m}, 98.3471\text{m}, 200\text{m}, 0.05\,\text{m/s}, -0.4067\,\text{m/s}, 0.05\,\text{m/s}\right]^T$, $\boldsymbol{\beta}_{c,0} = \left[1,1,1\right]^T$ deg/hour , and $\boldsymbol{\beta}_{d,0} = \left[1,1,1\right]^T$ deg/hour .

For $\mathbf{P}_0$, the initial covariance of the attitude is $\text{diag}\left(\left[20^\circ, 20^\circ, 20^\circ\right]^2\right)$. The initial covariance matrices of the gyro biases for the chief and the deputy are assumed to be the same and are $\text{diag}\left(\left[10^\circ/\text{h}, 10^\circ/\text{h}, 10^\circ/\text{h}\right]^2\right)$. The initial covariance of the position and

velocity are $\text{diag}\left(\left[10\text{m},10\text{m},10\text{m}\right]^2\right)$ and $\text{diag}\left(\left[0.01\text{m/s},0.01\text{m/s},0.01\text{m/s}\right]^2\right)$, respectively.

To achieve bounded relative orbit, the following constraint for the initial values should be satisfied [51]

$$\dot{y}_0 = -2\omega_r x_0 \tag{4.74}$$

and

$$y_0 = 2\dot{x}_0/\omega_r \tag{4.75}$$

where ω_r is given by Eq. (4.58).

The true angular velocities of the chief and deputy are given by $\boldsymbol{\omega}_c = \left[0, 0.001, -0.001\right]^T$ rad/s and $\boldsymbol{\omega}_d = \left[-0.001, 0, 0.001\right]^T$ rad/s, respectively. The standard deviations of the gyro biases for the chief and the deputy are $\sigma_{cu} = \sigma_{du} = \sqrt{10}\times 10^{-10}$ rad/s$^{3/2}$ and $\sigma_{cv} = \sigma_{dv} = \sqrt{10}\times 10^{-5}$ rad/s$^{3/2}$, respectively. Four beacons are used in the simulation. The locations of the beacons are listed in Table 4.5.

Table 4.5 Locations of beacons

	X_i	Y_i	Z_i
Beacon 1	0.5	0.5	0
Beacon 2	-0.5	0.5	0
Beacon 3	0.5	-0.5	0
Beacon 4	-0.5	-0.5	0

In the filters, the fourth-order Runge-Kutta method is used to propagate the state $\mathbf{x}_r$ of the relative orbit. The covariance in the measurement equation is $R_i = \left(0.0005^\circ\right)^2$.

The simulation time is 30 minutes and the sampling time is 1 second. The measurement period is 10 seconds. The following simulation results are based on 50 Monte Carlo runs. The averaged absolute estimation errors of each state are shown in Figure 4.19-4.23. From Figure 4.19, it can be seen that the SGHQFs and CKF have better performance than the EKF and the UKF ($\kappa = 3 - n = -12$) for attitude estimation. The perform of UKF is close to that of EKF and the level-2 SGHQF is close to CKF. Among all tested point-based Gaussian approximation filters, the level-3 SGHQF converges much faster than the UKF, the level-2 SGHQF, and the CKF, and achieves the best accuracy. Furthermore, from Figure 4.20 and Figure 4.21, only the level-3 SGHQF can obtain acceptable results for estimating the gyro biases of the chief and the deputy.

The relative position and velocity are shown in Figure 4.22 and CHAPTER IV, respectively, in which the EKF and UKF are not shown in because they cannot converge into acceptable ranges. It can be seen from Figure 4.22 and CHAPTER IV that the level-2 SGHQF and the CKF are very close and the level-3 SGHQF achieves the best performance in terms of the convergence rate and the estimation accuracy.

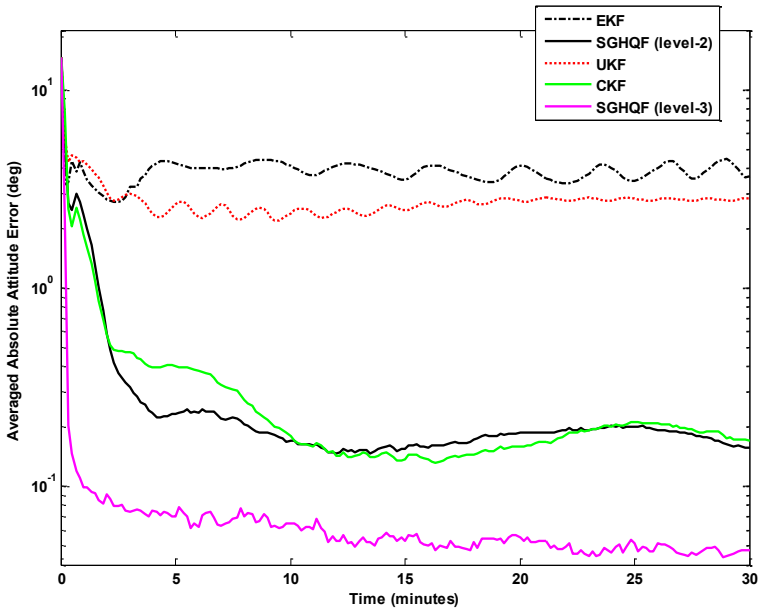

Figure 4.19 Averaged absolute attitude error of EKF, SGHQFs, UKF, and CKF

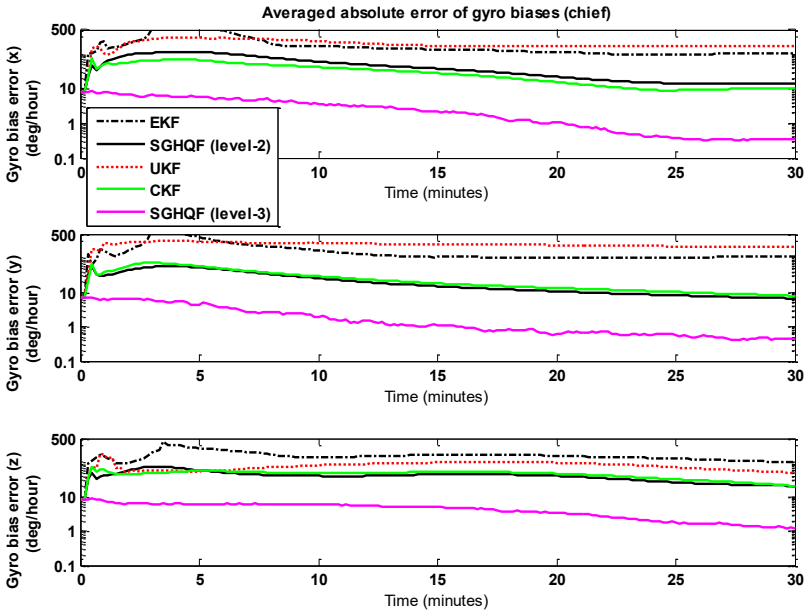

Figure 4.20 Averaged absolute error of gyro biases of EKF, SGHQFs, UKF, and CKF for the chief spacecraft

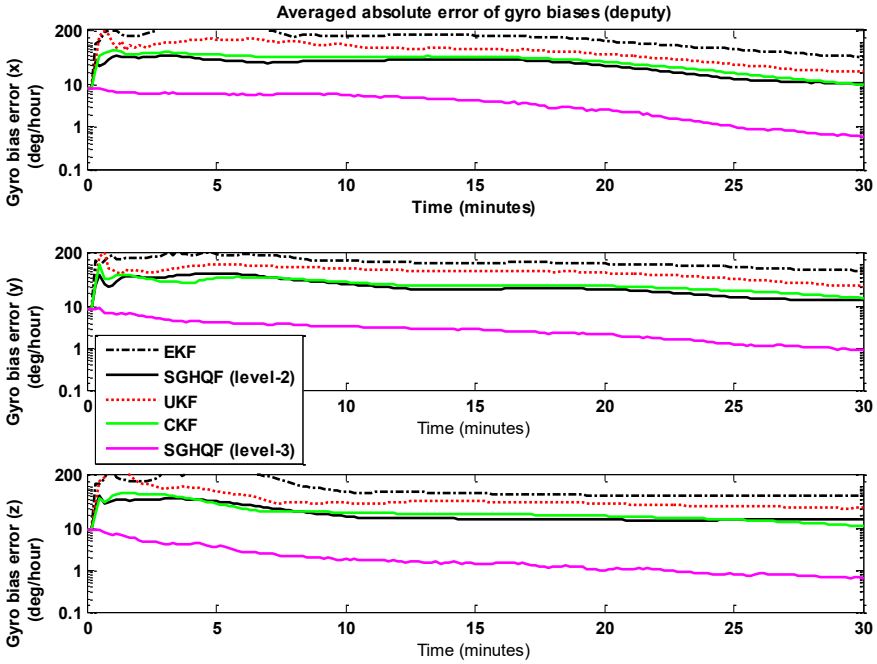

Figure 4.21 Averaged absolute error of gyro biases of EKF, SGHQFs, UKF, and CKF for the deputy spacecraft

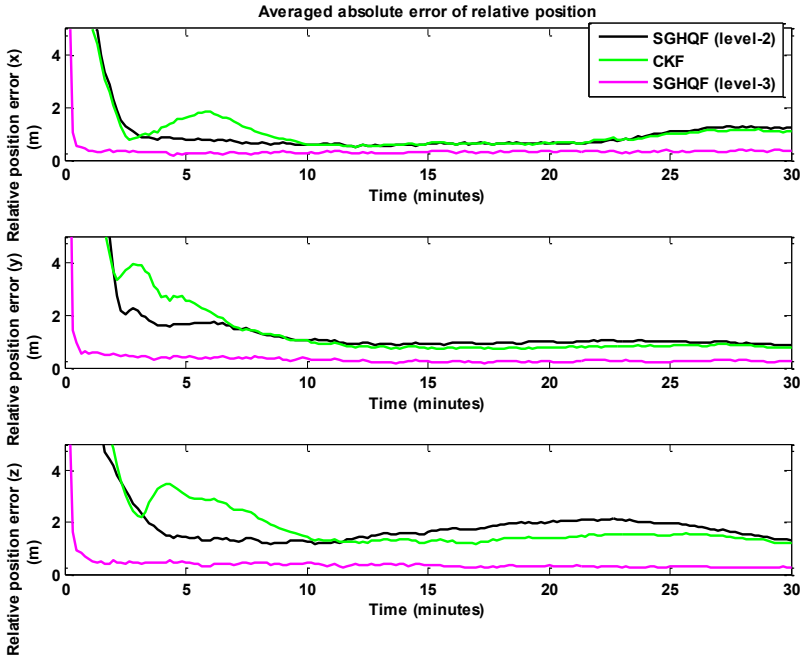

Figure 4.22 Averaged absolute error of relative position of SGHQFs and CKF

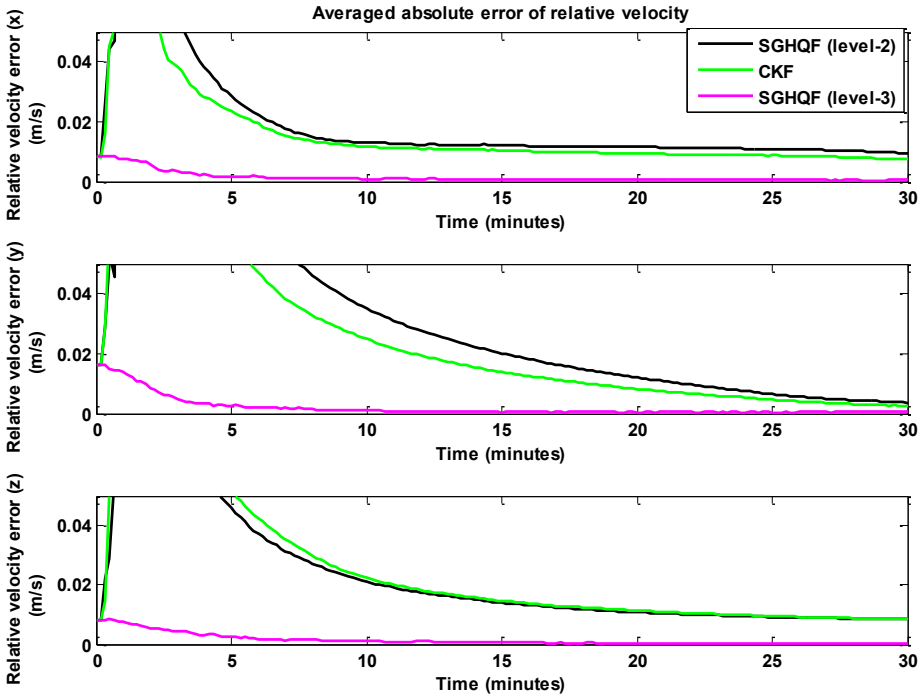

Figure 4.23 Averaged absolute error of relative velocity of SGHQFs and CKF

CHAPTER V

CONCLUSIONS AND FUTURE RESEARCH

To summarize this dissertation, the following conclusions can be obtained.

1) Three new sparse-grid based Gaussian approximation filters, including sparse Gauss-Hermite quadrature filter, sparse-grid quadrature filter, and anisotropic sparse-grid filter, have been proposed. The sparse Gauss-Hermite quadrature filter alleviates the *curse-of-dimensionality* problem of the conventional Gauss-Hermite quadrature filter while maintaining close performance to it. In addition, the sparse Gauss-Hermite quadrature filter can achieve higher accuracy than the extended Kalman filter and the unscented Kalman filter. The sparse-grid quadrature filter extended the sparse Gauss-Hermite quadrature filter by replacing the Gauss-Hermite quadrature rule with the moment matching method. Tunable parameters can be used to make the filtering design more flexible. The anisotropic sparse-grid quadrature filter provides an alternative to the sparse-grid quadrature filter to tradeoff the computation complexity with the estimation accuracy.

2) Theoretical analysis is given to prove the guaranteed performance of the three proposed sparse-grid based filters. It has been shown in this dissertation that the unscented Kalman filter with suggested parameter is a subset of the sparse Gauss-Hermite quadrature filter. In addition, the unscented Kalman filter with any parameter is a subset of the sparse-grid quadrature filter. The relationship among the numerical rules used in Gaussian approximation filters including the unscented transformation, the

cubature rule, the sparse Gauss-Hermite quadrature, the sparse-grid quadrature, and the anisotropic sparse-grid quadrature, has been established in terms of points, weights, and accuracy. It has been shown that all these point-based rules can be closely unified in the general sparse-grid framework. The sparse-grid based filters can achieve higher accuracy than the unscented Kalman filter. Furthermore, the computational complexity in terms of the number of points used in these filters has been analyzed. It is shown that the number of points increases polynomially with the increase of the dimension, which alleviates the *curse-of-dimensionality*.

3) Three aerospace applications, including spacecraft attitude estimation, orbit determination, and relative spacecraft navigation, are investigated to demonstrate the performance of the sparse-grid based filters. The simulation results have shown that the sparse-grid based filters achieve the best overall performance in terms of estimation accuracy and computational efficiency against the extended Kalman filter, unscented Kalman filter, cubature Kalman filter, and Gauss-Hermite quadrature filter.

There are two future research objectives. The first objective is to design a new sparse-grid filtering framework that can handle highly non-Gaussian systems. The sparse-grid generation in this dissertation will be extended to integrals with general probability density functions and new point propagation and update strategies will be developed. The second objective is to extend the sparse-grid filtering method to continuous-discrete systems, which can be used to solve the Fokker-Planck-Kolmogorov equation. The new filter will be based on the solution to the Fokker-Planck-Kolmogorov equation and the Bayesian update formula. The extended filter is expected to fully represent the probability density function and improve the accuracy of the conventional continuous-discrete Gaussian approximation filters.

REFERENCES

[1] P. M. Djuric, J. H. Kotecha, Z. Jianqui, H. Yufei, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *Signal Processing Magazine, IEEE,* vol. 20, pp. 19-38, 2003.

[2] M. R. Morelande, C. M. Kreucher, and K. Kastella, "A Bayesian approach to multiple target detection and tracking," *Signal Processing, IEEE Transactions on,* vol. 55, pp. 1589-1604, 2007.

[3] I. Bilik and J. Tabrikian, "Maneuvering target tracking in the presence of glint using the nonlinear Gaussian mixture Kalman filter," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 46, pp. 246-262, 2010.

[4] M. G. S. Bruno, "Bayesian methods for multiaspect target tracking in image sequences," *Signal Processing, IEEE Transactions on,* vol. 52, pp. 1848-1861, 2004.

[5] J. M. C. Clark, P. A. Kountouriotis, and R. B. Vinter, "A Gaussian mixture filter for range-only tracking," *Automatic Control, IEEE Transactions on,* vol. 56, pp. 602 -613, march 2011.

[6] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund, "Particle filters for positioning, navigation, and tracking," *Signal Processing, IEEE Transactions on,* vol. 50, pp. 425-437, 2002.

[7] P. Perez, J. Vermaak, and A. Blake, "Data fusion for visual tracking with particles," *Proceedings of the IEEE,* vol. 92, pp. 495-513, 2004.

[8] W. Chang, C. Chen, and Y. Hung, "Tracking by parts: A Bayesian approach with component collaboration," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 39, pp. 375-388, 2009.

[9] V. Fox, J. Hightower, L. Lin, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *Pervasive Computing, IEEE,* vol. 2, pp. 24-33, 2003.

[10] C. Yardim, Z. H. Michalopoulou, and P. Gerstoft, "An overview of sequential Bayesian filtering in ocean acoustics," *Oceanic Engineering, IEEE Journal of,* vol. 36, pp. 71-89, 2011.

[11] R. Sameni, M. B. Shamsollahi, C. Jutten, and G. D. Clifford, "A Nonlinear Bayesian filtering framework for ECG denoising," *Biomedical Engineering, IEEE Transactions on,* vol. 54, pp. 2172-2185, 2007.

[12] G. Ferraiuolo and G. Poggi, "A Bayesian filtering technique for SAR interferometric phase fields," *Image Processing, IEEE Transactions on,* vol. 13, pp. 1368-1378, 2004.

[13] J. L. Blanco, J. A. Fernandez-Madrigal, and J. Gonzalez, "Toward a unified Bayesian approach to hybrid metric--topological SLAM," *Robotics, IEEE Transactions on,* vol. 24, pp. 259-270, 2008.

[14] A. L. Juloski, S. Weiland, and W. P. M. H. Heemels, "A Bayesian approach to identification of hybrid systems," *Automatic Control, IEEE Transactions on,* vol. 50, pp. 1520-1533, 2005.

[15] J. Fang and X. Gong, "Predictive Iterated Kalman Filter for INS/GPS Integration and Its Application to SAR Motion Compensation," *Instrumentation and Measurement, IEEE Transactions on,* vol. 59, pp. 909-915, 2010.

[16] B. Brumback and M. Srinath, "A Chi-square test for fault-detection in Kalman filters," *Automatic Control, IEEE Transactions on,* vol. 32, pp. 552-554, 1987.

[17] F. N. Chowdhury, J. P. Christensen, and J. L. Aravena, "Power system fault detection and state estimation using Kalman filter with hypothesis testing," *Power Delivery, IEEE Transactions on,* vol. 6, pp. 1025-1030, 1991.

[18] G. R. Redinbo, "Generalized algorithm-based fault tolerance: error correction via Kalman estimation," *Computers, IEEE Transactions on,* vol. 47, pp. 639-655, 1998.

[19] A. Jazwinski, *Stochastic Processing and Filtering Theory*: Academic Press, New York, NY, 1970.

[20] S. Challa and Y. Bar-Shalom, "Nonlinear filter design using Fokker-Planck-Kolmogorov probability density evolutions," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 36, pp. 309-315, 2000.

[21] S. Challa, Y. Bar-Shalom, and V. Krishnamurthy, "Nonlinear filtering via generalized Edgeworth series and Gauss-Hermite quadrature," *Signal Processing, IEEE Transactions on,* vol. 48, pp. 1816-1820, 2000.

[22] F. Daum, "Nonlinear filters: beyond the Kalman filter," *Aerospace and Electronic Systems Magazine, IEEE,* vol. 20, pp. 57-69, 2005.

[23] S. Chakravorty, M. Kumar, and P. Singla, "A quasi-Gaussian Kalman filter," in *American Control Conference, 2006*, 2006, pp. 970-975.

[24] Y. Xu and P. Vedula, "A quadrature-based method of moments for nonlinear filtering," *Automatica,* vol. 45, pp. 1291-1298, 2009.

[25] 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, pp. 367-379, 2011.

[26] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEEE Proceedings,* vol. 140, pp. 107-113, Apr 1993.

[27] M. Simandl, J. Kralovec, and T. Soderstrom, "Anticipative grid design in point-mass approach to nonlinear state estimation," *Automatic Control, IEEE Transactions on,* vol. 47, pp. 699-702, 2002.

[28] M. Šimandl, J. Královec, and T. Söderström, "Advanced point-mass method for nonlinear state estimation," *Automatica,* vol. 42, pp. 1133-1145, 2006.

[29] I. Arasaratnam, S. Haykin, and R. J. Elliott, "Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature," *Proceedings of the IEEE,* vol. 95, pp. 953-977, 2007.

[30] V. Klumpp, F. Sawo, U. D. Hanebeck, and D. Franken, "The sliced Gaussian mixture filter for efficient nonlinear estimation," in *Information Fusion, 2008 11th International Conference on*, ed, 2008, pp. 1 -8.

[31] S. S. Ali- ytty, ox Gaussian mixture filter, *Automatic Control, IEEE Transactions on,* vol. 55, pp. 2165-2169, 2010.

[32] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on,* vol. 50, pp. 174-188, Feb 2002.

[33] X. Luo, I. M. Moroz, and I. Hoteit, "Scaled unscented transform Gaussian sum filter: Theory and application," *Physica D: Nonlinear Phenomena,* vol. 239, pp. 684-701, 2010.

[34] Z. Ye and Y. Zhang, "Speed estimation from single loop data using an unscented particle filter," *Computer-Aided Civil and Infrastructure Engineering,* vol. 25, pp. 494-503, 2010.

[35] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control,* vol. 45, pp. 477-482, 2000.

[36] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE,* vol. 92, pp. 401-422, 2004.

[37] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *Automatic Control, IEEE Transactions on,* vol. 54, pp. 1254-1269, 2009.

[38] R. Cools and P. Rabinowitz, "Monomial cubature rules since "Stroud": a compilation," *Journal of Computational and Applied Mathematics,* vol. 48, pp. 309-326, 1993.

[39] C. Ronald, "An encyclopaedia of cubature formulas," *Journal of Complexity,* vol. 19, pp. 445-453, 2003.

[40] C. Ronald, Monomial cubature rules since "Stroud": a compilation — part 2," *Journal of Computational and Applied Mathematics,* vol. 112, pp. 21-27, 1999.

[41] S. A.Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Soviet. Math. Dokl. 4,* pp. 240-243, 1963.

[42] A. Gelb, *Applied Optimal Estimation*: The MIT Press, 1974.

[43] A. Carmi and Y. Oshman, "Fast particle filtering for attitude and angular-rate estimation from vector observations," *Journal of Guidance, Control, and Dynamics,* vol. 32, pp. 70-78, 2009.

[44] Y. Cheng and J. L. Crassidis, "Particle filtering for attitude estimation using a minimal local-error representation," *Journal of Guidance, Control, and Dynamics,* vol. 33, pp. 1305-1310, 2010.

[45] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *Signal Processing, IEEE Transactions on,* vol. 51, pp. 2602 - 2612, oct. 2003.

[46] Y. Oshman and A. Carmi, "Attitude estimation from vector observations using Genetic-Algorithm-Embedded quaternion particle filter," *Journal of Guidance, Control, and Dynamics,* vol. 29, pp. 879-891, 2006.

[47] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics,* vol. 6, pp. 419-430, 1982.

[48] M. D. Shuster, "A simple Kalman filter and smoother for spacecraft attitude," *Journal of the Astronautical Sciences,* vol. 37, pp. 89-106, 1989.

[49] M. L. Psiaki, "Attitude-determination filtering via extended quaternion estimation," *Journal of Guidance, Control, and Dynamics,* vol. 23, pp. 206-214, 2000.

[50] J. L. Crassidis and F. L. Markley, "Unscented filtering for spacecraft attitude estimation," *Journal of Guidance, Control, and Dynamics,* vol. 26, pp. 536-542, Aug.2003.

[51] D. A. Vallado, *Fundamentals of Astrodynamics and Applications*: Microcosm Press and Kluwer Academic Publishers, 2001.

[52] D.-J. Lee and K. T. Alfriend, "Sigma point filtering for sequential orbit estimation and prediction," *Journal of Spacecraft and Rockets,* vol. 44, pp. 388-398, March-April 2007.

[53] G. Wang and X.-j. Duan, "Particle filtering and its application in satellite orbit determination," in *Image and Signal Processing, 2008. CISP '08. Congress on* vol. 5, ed, 2008, pp. 488-492.

[54] N. K. Philip and M. R. Ananthasayanam, "Relative position and attitude estimation and control schemes for the final phase of an autonomous docking mission of spacecraft," *Acta Astronautica,* vol. 52, pp. 511 - 522, 2003.

[55] Y. Xing, X. Cao, S. Zhang, H. Guo, and F. Wang, "Relative position and attitude estimation for satellite formation with coupled translational and rotational dynamics," *Acta Astronautica,* vol. 67, pp. 455 - 467, 2010.

[56] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *Automatic Control, IEEE Transactions on,* vol. 45, pp. 910-927, 2000.

[57] M. Šimandl and J. Duník, Derivative-free estimation methods: New results and performance analysis," *Automatica,* vol. 45, pp. 1749-1757, 2009.

[58] A. H. Stroud., *Approximate Calculation of Multiple Integrals*. NJ: Prentice-Hall, Englewood Cliffs, 1971.

[59] Y. Wu, D. Hu, M. Wu, and X. Hu, "A numerical-integration perspective on Gaussian filters," *Signal Processing, IEEE Transactions on,* vol. 54, pp. 2910-2921, 2006.

[60] S. J. Julier and J. K. Uhlmann, "Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations," in *American Control Conference, 2002. Proceedings of the 2002*, 2002, pp. 887-892 vol.2.

[61] S. J. Julier, "The spherical simplex unscented transformation," in *American Control Conference, 2003. Proceedings of the 2003*, 2003, pp. 2430-2434 vol.3.

[62] J. G. C. Lozano, L. R. G. Carrillo, A. Dzul, and R. Lozano, "Spherical simplex sigma-point Kalman filters: A comparison in the inertial navigation of a terrestrial vehicle," in *American Control Conference, 2008*, 2008, pp. 3536-3541.

[63] W. Li, P. Wei, and X. Xiao, "A novel simplex unscented transform and filter," in *Communications and Information Technologies, 2007. ISCIT '07. International Symposium on*, 2007, pp. 926-931.

[64] J. F. Levesque, "Second-order simplex sigma points for nonlinar estimaton," presented at the AIAA Guidiance, Navigation, and Control Conference, Keystone, Colorado, 2006.

[65] J. R. V. Zandt, "A more robust unscented transform," MITRE Corporation2001.

[66] F. Sun, Y. Ma, and J. Wang, "Extended symmetric sampling strategy for unscented Kalman filter," in *Control, Automation and Systems Engineering, 2009. CASE 2009. IITA International Conference on*, 2009, pp. 383-386.

[67] W. Zhang, M. Liu, and Z. Zhao, "Accuracy analysis of unscented transformation of several sampling strategies," in *Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 2009. SNPD '09. 10th ACIS International Conference on*, 2009, pp. 377-380.

[68] J. Dunik, M. Simandl, O. Straka, and L. Kral, "Performance analysis of derivative-free filters," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, 2005, pp. 1941-1946.

[69] J. Lu and D. L. Darmofal, "Higher-dimensional integration with Gaussian weight for applications in probabilistic design," *SIAM Journal on Scientific Computing,* vol. 26, pp. 613-624, 2004.

[70] A. Genz and J. Monahan, "A stochastic algorithm for high-dimensional integrals over unbounded regions with Gaussian weight," *Journal of Computational and Applied Mathematics,* vol. 112, pp. 71-81, 1999.

[71] M. Nørgaard, N. K. Poulsen, and O. Ravn, "New developments in state estimation for nonlinear systems," *Automatica,* vol. 36, pp. 1627-1638, 2000.

[72] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *Signal Processing, IEEE Transactions on,* vol. 51, pp. 2602-2612, 2003.

[73] P. J. Nordlund and F. Gustafsson, "Marginalized particle filter for accurate and reliable terrain-aided navigation," *Aerospace and Electronic Systems, IEEE Transactions on,* vol. 45, pp. 1385-1399, 2009.

[74] U. Orguner and F. Gustafsson, "Risk-sensitive particle filters for mitigating sample impoverishment," *Signal Processing, IEEE Transactions on,* vol. 56, pp. 5001-5012, 2008.

[75] T. Schon, F. Gustafsson, and P. J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *Signal Processing, IEEE Transactions on,* vol. 53, pp. 2279-2289, 2005.

[76] S. C. Kramer and H. W. Sorenson, "Recursive Bayesian estimation using piece-wise constant approximations," *Automatica,* vol. 24, pp. 789-801, 1988.

[77] J. T. Horwood and A. B. Poore, "Adaptive Gaussian sum filters for space surveillance," *Automatic Control, IEEE Transactions on,* vol. 56, pp. 1777-1790, 2011.

[78] J. Yin, J. Zhang, and Z. Zhuang, "Gaussian sum PHD filtering algorithm for nonlinear Non-Gaussian models," *Chinese Journal of Aeronautics,* vol. 21, pp. 341-351, 2008.

[79] N. M. Kwok, Q. P. Ha, S. Huang, G. Dissanayake, and G. Fang, "Mobile robot localization and mapping using a Gaussian sum filter," *International Journal of Control, Automation and Systems,* vol. 5, pp. 251-268, 2007.

[80] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "Adaptive Gaussian sum filter for nonlinear Bayesian estimation," *Automatic Control, IEEE Transactions on,* vol. 56, pp. 2151-2156, 2011.

[81] O. Cappe, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE,* vol. 95, pp. 899-924, 2007.

[82] X. Deng, J. Xie, and H. Ni, "Interacting multiple model algorithm with the unscented particle filter (UPF)," *Chinese Journal of Aeronautics,* vol. 18, pp. 366-371, 2005.

[83] Z. Liang, X. Ma, and X. Dai, "Robust tracking of moving sound source using scaled unscented particle filter," *Applied Acoustics,* vol. 69, pp. 673-680, 2008.

[84] F. Nobile, R. Tempone, and C. G. Webster, "A Sparse grid stochastic collocation method for partial differential equations with random input data," *SIAM J. Numer. Anal.,* vol. 46, pp. 2309-2345, May 2008.

[85] F. Heiss and V. Winschel, "Likelihood approximation by numerical integration on sparse grids," *Journal of Econometrics,* vol. 144, pp. 62-80, May 2008.

[86] G. W. Wasilkowski and H. Wozniakowski, "Explicit cost bounds of algorithms for multivariate tensor product problems," *Journal of Complexity,* vol. 11, pp. 1-56, 1995.

[87] N. Macon and A. Spitzbart, "Inverses of vandermonde matrices," *The American Mathematical Monthly,* vol. 65, pp. pp. 95-100, 1958.

[88] D. Tenne and T. Singh, "The higher order unscented filter," in *American Control Conference, 2003. Proceedings of the 2003*, 2003, pp. 2441-2446 vol.3.

[89] F. Nobile, R. Tempone, and C. G. Webster, "An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data," *SIAM J. Numer. Anal.,* vol. 46, pp. 2411-2442, June 2008.

[90] J. L. Crassidis, F. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of Guidance, Control, and Dynamics,* vol. 30, pp. 12-28, 2007.

[91] J. L. Crassidis. and J. L. Junkins., *Optimal Estimation of Dynamic Systems*: Chapman and Hall/CRC, 2004.

[92] J. E. Prussing and B. A. Conway, *Orbital Mechanics*. New York, NY: Oxford University Press, 1993.

[93] S.-G. Kim, J. L. Crassidis, Y. Cheng, and A. M. Fosbury, "Kalman filtering for relative spacecraft attitude and position estimation," *Journal Of Guidance, Control, And Dynamics,* vol. 30, pp. 133-143, 2007.

[94] D. Xue, X. Cao, and Y. Wu, "Decentralized determination of relative orbit for formation flying satellite," in *Systems and Control in Aerospace and Astronautics, 2006. ISSCAA 2006. 1st International Symposium on*, 2006, pp. 338-343.

[95] M. L. Psiaki. and S. Mohiuddin., "Modeling, analysis, and simulation of GPS carrier phase for spacecraft relative navigation," *Journal of Guidance, Control, and Dynamics,* vol. 30, pp. pp. 1628-1639, 2007.

[96] S. Mohiuddin and M. L. Psiaki, "High-altitude satellite relative navigation using carrier-phase differential global positioning system techniques," *Journal of Guidance, Control, and Dynamics,* vol. 30, pp. pp. 1427-1436, 2007.

[97] Z. Dang and Y. Zhang, "Relative position and attitude estimation for Inner-formation gravity measurement satellite system," *Acta Astronautica,* vol. 69, pp. 514-525, 2011.

[98] X. Wang, D. Gong, L. Xu, X. Shao, and D. Duan, "Laser radar based relative navigation using improved adaptive Huber filter," *Acta Astronautica,* vol. 68, pp. 1872-1880, 2011.

[99] T. Chen and S. Xu, "Double line-of-sight measuring relative navigation for spacecraft autonomous rendezvous," *Acta Astronautica,* vol. 67, pp. 122-134, 2009.

[100] K. K. Gunnam, D. C. Hughes, J. L. Junkins, and N. Kehtarnavaz, "A vision-based DSP embedded navigation sensor," *Sensors Journal, IEEE,* vol. 2, pp. 428-442, 2002.

[101] D. Lee. and H. Pernicka., "Vision-based relative state estimation using the unscented Kalman filter," *International Journal of Aeronautical and Space Sciences,* vol. 12, pp. pp. 24-36, 2011.