Journal of Systems Engineering and Electronics
Vol. 28, No. 3, June 2017, pp.556 – 562



# Generalized cubature quadrature Kalman filters: derivations and extensions

Hongwei Wang[1,3], Wei Zhang[1,2,*], Junyi Zuo[1,2], and Heping Wang[1]

1. School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China;
2. Experimental Aircraft Design and Flight Testing Laboratory of Shaanxi, Xi'an 710072, China;
3. School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore

**Abstract:** A new Gaussian approximation nonlinear filter called generalized cubature quadrature Kalman filter (GCQKF) is introduced for nonlinear dynamic systems. Based on standard GCQKF, two extensions are developed, namely square root generalized cubature quadrature Kalman filter (SR-GCQKF) and iterated generalized cubature quadrature Kalman filter (I-GCQKF). In SR-GCQKF, the QR decomposition is exploited to alter the Cholesky decomposition and both predicted and filtered error covariances have been propagated in square root format to make sure the numerical stability. In I-GCQKF, the measurement update step is executed iteratively to make full use of the latest measurement and a new terminal criterion is adopted to guarantee the increase of likelihood. Detailed numerical experiments demonstrate the superior performance on both tracking stability and estimation accuracy of I-GCQKF and SR-GCQKF compared with GCQKF.

**Keywords:** cubature rule, quadrature rule, Kalman filter, iterated method, QR decomposition, nonlinear estimation, target tracking.

**DOI:** 10.21629/JSEE.2017.03.15

## 1. Introduction

Nonlinear filtering or state estimation based on Bayesian inference has been an active research area and has been widely used in various fields, such as navigation, target tracking, system identification, fault detection and isolation, multi-source information fusion, statistic signal processing, and econometrics [1 – 5]. Unlike the celebrated Kalman filter [6], nonlinear filtering can rarely be expressed in close-form due to the existence of intractable multivariable integrals, thus many researchers have to abandon the idea of obtaining analytic solutions and tend to seek suboptimal solutions [7].

Two main approaches appear in literature to reach sub-optimal solution [8], i.e., global approach and local approach. A set of weighted particles are randomly selected to approximate the propagation of possible density functions (PDFs) in the global approach. Representative cases in the global approximation are particle filter [9] and grids point-mass filter [10]. Although filters in this approach have high precision, it is their computational burden that makes them far from meeting the real time request in the practical applications.

In local approximations, one intuitive scheme is to successively linearize state process and observation mapping, giving birth to the extended Kalman filter (EKF) [12]. However, its inherent deficiencies limit the estimation accuracy. Another scheme is called the deterministic sampling filter, in which a certain Gaussian PDF is selected to approximate the non-Gaussian PDFs naturally caused by the nonlinearity in some sense. Unscented Kalman filter (UKF) [13] and Gauss-Hermite quadrature Kalman filter (GHQF) [15] are well-known representatives of this scheme.

Recently a kind of novel local-approach filter has been developed which shares the same foundational theories, namely, spherical radical cubature rule and Gauss-Laguerre quadrature rule. The cubature Kalman filter (CKF) [16] exploited the third degree spherical-radial cubature rule, having better estimation results than UKF while less computational burden than GHKF. Then for further improving estimation accuracy, high-degree CKF [17], cubature quadrature Kalman filter (CQKF) [18] and high degree CQKF [19] have been proposed. In this paper, we summarize the former work and derive the generalized cubature quadrature Kalman filters (GCQKF). Two parameters, i.e., degree of cubature rule and order of quadrature rule, should be determined in GCQKF since

Manuscript received December 10, 2015.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61473227; 11472222), the Aerospace Technology Support Fund of China (2014-HT-XGD), the Natural Science Foundation of Shaanxi Province (2015JM6304) and the Aeronautical Science Foundation of China (20151353018).

different parameters lead to different filters.

All GCQKFs work well in ideal situations. However, in real applications, some physical restrictions such as limited computing length may lead to truncation error and induce the loss of semi-positive character of the covariance matrix, which may result in failure of filter tracking. In addition, in some scenarios such a ballistic target tracking the significant initial error is inevitable and as a result, the loss of tracking accuracy may arise. For sake of better performance when dealing with real problems with large initial error or limited computing length, we propose two extensions of GCQKF, namely square root GCQKF (SR-GCQKF) and iterated GCQKF (I-GCQKF).

## 2. Derivations of GCQKF

In Bayesian filtering for discrete-time dynamical systems (DDS), the posterior density, which provides complete statistical information of state, can be calculated recursively by two steps: prediction update and measurement update, governed by the Chapman-Kolmogorov equation and Bayes rule respectively [16]. Generally speaking, both steps are intractable due to the involvement of multivariate integrals. Under the assumption that all conditional PDFs are Gaussian, the Bayesian filter is rendered tractable to implement since all related integrals are turned to multi-dimensional Gaussian weighted integrals. The general format of the Gaussian approximation filter can be found in [14].

It is known that for arbitrary function $f(\boldsymbol{x}), \boldsymbol{x} \in \mathbf{R}^n$, the Gaussian-weighted integral

$$I(f) = \frac{1}{\sqrt{|\boldsymbol{\Sigma}|\,(2\pi)^n}} \int_{\mathbf{R}^n} f(\boldsymbol{x}) \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \mathrm{d}\boldsymbol{x} \tag{1}$$

can be expressed as

$$I(f) = \frac{1}{\sqrt{(2\pi)^n}} \int_{r=0}^{\infty} \int_{U_n} [f(\boldsymbol{x})\mathrm{d}s(\boldsymbol{z})]\, r^{n-1} \mathrm{e}^{-\frac{r^2}{2}} \mathrm{d}r \tag{2}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance of a Gaussian distribution $p(\boldsymbol{x})$, respectively; $\boldsymbol{x} = \boldsymbol{C}r\boldsymbol{z} + \boldsymbol{\mu}$ is an affine transformation in which $\boldsymbol{C}$ is the Cholesky decomposition of $\boldsymbol{\Sigma}$; $U_n$ is the surface of a unit hypersphere definded by $U_n = \{\boldsymbol{z} \in \mathbf{R}^n | \boldsymbol{z}^{\mathrm{T}}\boldsymbol{z} = 1\}$; $s(\boldsymbol{z})$ is the area element on $U_n$. That is, the Gaussian-weighted integrals can be decomposed into two tractable parts: a surface integral and a line integral. The spherical cubature rule can be applied to calculate the surface integral over a hyper-sphere and the Gauss-Laguerre quadrature rule can be used to evaluate the line integral.

### 2.1 Spherical cubature rule

A variety of spherical cubature rules were introduced in [20] to solve the surface integral and all those cubature rules can be exploited to deal with surface integral in GC-QKF. For definiteness and without loss of generality, we here adopt the most commonly used cubature rule called Genzs method, which was first proposed by Genz [21] based on Silvesters integration rules [22].

For a surface integral $I_{U_n}(f) \triangleq \int_{U_n} f(s)\mathrm{d}\sigma(s)$, it can be calculated numerically by the $(2m+1)$th degree cubature rule as

$$I_{U_n}(f) \approx \sum_{|c|=m} \omega_c f\{\boldsymbol{u}_c\} \tag{3}$$

where the cubature points set $\{\boldsymbol{u}_c\}$ and corresponding weight $\omega_c$ are given by

$$\{\boldsymbol{u}_{\boldsymbol{c}}\} \triangleq \bigcup \{\beta_1 \boldsymbol{u}_{c_1}, \beta_2 \boldsymbol{u}_{c_2}, \ldots, \beta_n \boldsymbol{u}_{c_n}\}$$

$$\omega_c \triangleq 2^{-o(\boldsymbol{u}_c)} I_{U_n} \left( \prod_{i=1}^{n} \prod_{j=0}^{c_i-1} \frac{\boldsymbol{s}_i^2 - \boldsymbol{u}_j^2}{\boldsymbol{u}_{c_i}^2 - \boldsymbol{u}_j^2} \right)$$

where $c$ is a set of non-negative numbers with $\boldsymbol{c} = [c_1, c_2, \ldots, c_n]$ which satisfies $|\boldsymbol{c}| = \sum_{i=1}^{n} c_i = m$; $\beta_i = \pm 1$ and $u_{c_i} = \sqrt{c_i/m}$; the superscript $o(\boldsymbol{u}_c)$ means the number of nonzero elements in $\boldsymbol{u}_c$.

### 2.2 Gauss-Laguerre quadrature rule

Line integral from 0 to infinite can be approximated by the generalized Gauss-Laguerre quadrature rule as

$$I(f) = \int_0^{\infty} f(\lambda)\lambda^{\alpha} e^{-\lambda} d\lambda \approx \sum_{i}^{n_q} \omega_i \lambda_i \tag{4}$$

where the $n_q$th order quadrature points $\lambda_i (i = 1, \ldots, n_q)$ are obtained by solving the following Chebyshey-Laguerre polynomial equation

$$L_{n_q}^{\alpha}(\lambda) = (-1)^{n_q} \lambda^{\alpha} \mathrm{e}^{-\lambda} \frac{\mathrm{d}^{n_q}}{\mathrm{d}\lambda^{n_q}} \lambda^{\alpha+n_q} \mathrm{e}^{-\lambda} = 0$$

and the corresponding weight for $\lambda_i$ is determined as

$$\omega_i = \frac{n_q! \Gamma(\alpha + n_q + 1)}{\lambda_i [(L_{n_q}^{\alpha}(\lambda_i))']^2}.$$

### 2.3 Arbitrary cubature quadrature rule

Using the spherical cubature rule and the Gauss-Laguerre quadrature rule, the integral described in (2) can be evaluated as

$$I(f) \approx \frac{1}{2\sqrt{\pi^n}} \sum_{i=1}^{n_q} \sum_{j=1}^{n_c} \omega_i \omega_j f(\sqrt{2\lambda_i}\boldsymbol{u}_j) \tag{5}$$

where $n_c$ is the number of cubature points which is a function with respect to the degree of cubature rule and $n_q$ is

the number of quadrature points which is the same as order of quadrature rule. We name the new point set as cubature quadrature point (CQ point). The complete algorithm to generate CQ point is summarized as follows. Readers can refer to [19] to find some examples about how to generate CQ points and their weights.

**Algorithm for CQ points generation**

**Step 1** Initialize the state dimension, degree of cubature rule and quadrature rule;

**Step 2** Find the cubature point set $\{\boldsymbol{u}_c^{(a)}, \omega_c^{(a)}\}_{a=1}^{n_c}$;

**Step 3** Find the quadrature point set $\{\lambda^{(b)}, \omega_q^{(b)}\}_{b=1}^{n_q}$;

**Step 4** Obtain the CQ point set as $\{\boldsymbol{\xi}_j, \omega_{cq}^{(j)}\}_{j=1}^{n_{cq}}$, where $n_{cq} = n_c n_q$, $\boldsymbol{\xi}_j = \sqrt{2\lambda_b}u_a$, $\omega_{cq}^{(j)} = \dfrac{\omega_c^{(a)}\omega_q^{(b)}}{2\sqrt{\pi^n}}$.

### 2.4 Implementation of GCQKF

Consider the following nonlinear DDS with additive noise:

$$\begin{cases} \boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}) + \boldsymbol{\nu}_{k-1} \\ \boldsymbol{y}_k = h(\boldsymbol{x}_k) + \boldsymbol{\omega}_k \end{cases} \tag{6}$$

where $\boldsymbol{x}_k \in \mathbf{R}^n$ is the state of interests at the time instant $k$; $f(\cdot)$ and $h(\cdot)$ are some known nonlinear functions; $\boldsymbol{y} \in \mathbf{R}^m$ is the measurement; $\boldsymbol{\mu}_{k-1}$ and $\boldsymbol{\omega}_k$ are independent Gaussian noise with zero means and covariance with $\boldsymbol{Q}_{k-1}$ and $\boldsymbol{R}_k$ respectively.

Before getting started, initialize the mean and covariance for random variable $\boldsymbol{x}_0$ to be $\hat{\boldsymbol{x}}_{0|0}$ and $\boldsymbol{P}_{0|0}$ respectively. Also the degree of cubature rule and quadrature rule should be determined and corresponding CQ points should be calculated as $\{\boldsymbol{\xi}_j, \omega_{cq}^{(j)}\}_{j=1}^{n_{cq}}$.

**Prediction step** Compute the Cholesky decomposition $\boldsymbol{P}_{k-1|k-1} = \boldsymbol{S}_{k-1|k-1}\boldsymbol{S}_{k-1|k-1}^{\mathrm{T}}$ and evaluate the cubature quadrature points as $\boldsymbol{\chi}_j = \boldsymbol{S}_{k-1|k-1}\boldsymbol{\xi}_j + \hat{\boldsymbol{x}}_{k-1|k-1}$. Update the prediction density $p_{k|k-1} = \mathcal{N}(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{P}_{k|k-1})$ by

$$\begin{cases} \hat{\boldsymbol{x}}_{k|k-1} = \displaystyle\sum_{j=1}^{n_{cq}} \omega_{cq}^{(j)} f(\boldsymbol{\chi}_j) \\ \boldsymbol{G} = \displaystyle\sum_{j=1}^{n_{cq}} \omega_{cq}^{(j)} (f(\boldsymbol{\chi}_j) - \hat{\boldsymbol{x}}_{k|k-1})(f(\boldsymbol{\chi}_j) - \hat{\boldsymbol{x}}_{k|k-1})^{\mathrm{T}} \\ \boldsymbol{P}_{k|k-1} = \boldsymbol{G} + \boldsymbol{Q}_{k-1} \end{cases} \tag{7}$$

**Measurement update step** Compute the factorization $\boldsymbol{P}_{k|k-1} = \boldsymbol{S}_{k|k-1}\boldsymbol{S}_{k|k-1}^{\mathrm{T}}$ and set new CQ points as $\boldsymbol{\chi}_j' = \boldsymbol{S}_{k|k-1}\boldsymbol{\xi}_j + \hat{\boldsymbol{x}}_{k|k-1}$. Update filtering density $p_{k|k} = \mathcal{N}(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{P}_{x|x})$ by

$$\begin{cases} \hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k) \\ \boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{P}_{yy}\boldsymbol{K}_k^{\mathrm{T}} \end{cases} \tag{8}$$

where

$$\hat{\boldsymbol{y}}_k = \sum_{j=1}^{n_{cq}} \omega_{cq}^{(j)} h(\boldsymbol{\chi}_j')$$

$$\boldsymbol{P}_{yy} = \sum_{j=1}^{n_{cq}} \omega_{cq}^{(j)} (h(\boldsymbol{\chi}_j') - \hat{\boldsymbol{y}}_k)(h(\boldsymbol{\chi}_j') - \hat{\boldsymbol{y}}_k)^{\mathrm{T}} + \boldsymbol{R}_k$$

$$\boldsymbol{P}_{xy} = \sum_{j=1}^{n_{cq}} \omega_{cq}^{(j)} (\boldsymbol{\chi}_j' - \hat{\boldsymbol{x}}_{k|k-1})(h(\boldsymbol{\chi}_j') - \hat{\boldsymbol{y}}_k)^{\mathrm{T}}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{xy}\boldsymbol{P}_{yy}^{-1}.$$

The number of CQ points is an important design parameter for GCQKF because it affects the computational efficiency directly. The number of CQ points is proportional to the order of quadrature rule while dramatically increasing with the growth of degree of cubature rules. Hence when designing the GCQKF, one should carefully choose the degree of cubature rule and order of quadrature rule to balance the estimation accuracy and computational burden.

## 3. Extensions of GCQKF

### 3.1 Square root GCQKF

As can be seen, the Cholesky decomposition is executed twice in each recursion step in GCQKF. In the Cholesky decomposition the matrix should be positively definited, otherwise the filter would be unstable and divergent. In general, however, the real applications may not meet the demand due to the truncation error of digital computers. As an alteration, orthogonal triangular decomposition, or QR decomposition is a promising approach. In addition, using the QR decomposition brings other benefits such as numerical accuracy improvement and storage space decrease [23].

In GCQKF the error covariance $\boldsymbol{P}$ is defined as $\boldsymbol{P} = \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$, where $\boldsymbol{X} \in \mathbf{R}^{n\times n_{cq}}$ for the predicted error covariance while $\boldsymbol{X} \in \mathbf{R}^{m\times n_{cq}}$ for the filtered error covariance. Apply the QR decomposition to $\boldsymbol{X}^{\mathrm{T}}$, then we get $\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{Q} \in \mathbf{R}^{n_{cq}\times n_{cq}}$ is an orthogonal matrix and $\boldsymbol{R} \in \mathbf{R}^{n_{cq}\times n}$ (or $\boldsymbol{R} \in \mathbf{R}^{n_{cq}\times m}$) is an upper triangular matrix. Hence $\boldsymbol{P}$ can be rewritten as

$$\boldsymbol{P} = \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{R} = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{R} = \boldsymbol{S}\boldsymbol{S}^{\mathrm{T}}$$

where $\boldsymbol{S} = \boldsymbol{R}^{\mathrm{T}}$. For brevity, we use $uptria(\boldsymbol{X})$ to represent $\boldsymbol{R}$ in the QR decomposition for $\boldsymbol{X}$.

Similarly, before starting filtering, we set the initial values $\hat{\boldsymbol{x}}_{0|0}$ and $\boldsymbol{P}_{0|0} = \boldsymbol{S}_{0|0}\boldsymbol{S}_{0|0}^{\mathrm{T}}$. Then we calculate the square root of noise covariance, i.e., $\boldsymbol{S}_{Q_k} = \sqrt{\boldsymbol{Q}_k}$ and $\boldsymbol{S}_{R_k} = \sqrt{\boldsymbol{R}_k}$. We also determine the CQ points $\{\boldsymbol{\xi}_j, \omega_{cq}^{(j)}\}_{j=1}^{n_{cq}}$ and the square weight matrix $\boldsymbol{\Pi} = \mathrm{diag}(\sqrt{\omega_{cq}^{(1)}}, \ldots, \sqrt{\omega_{cq}^{(n_{cq})}})$.

**Prediction step** Obtain the cubature quadrature points as $\boldsymbol{\chi}_j = \boldsymbol{S}_{k-1|k-1}\boldsymbol{\xi}_j + \hat{\boldsymbol{x}}_{k-1|k-1}$ and update the prediction mean $\hat{\boldsymbol{x}}_{k|k-1} = \sum_{j=1}^{n_{cq}} \omega_{cq}^{(j)} f(\boldsymbol{\chi}_j)$. Perform the QR decomposition $\boldsymbol{S}_{k|k-1} = uptria([\boldsymbol{X}_{k|k-1}, \boldsymbol{S}_{Q_{k-1}}])$, where $\boldsymbol{X}_{k|k-1}$ is defined as $\boldsymbol{X}_{k|k-1} = [f(\boldsymbol{\chi}_1) - \hat{\boldsymbol{x}}_{k|k-1}\ ,\ \ldots\ ,\ f(\boldsymbol{\chi}_{n_{cq}}) - \hat{\boldsymbol{x}}_{k|k-1}]\boldsymbol{\Pi}$.

**Measurement update step** Evaluate the CQ points as $\boldsymbol{\chi}'_j = \boldsymbol{S}_{k|k-1}\boldsymbol{\xi}_j + \hat{\boldsymbol{x}}_{k|k-1}$. Calculate the predicted measurement mean $\hat{\boldsymbol{y}}_k = \sum_{j=1}^{n_{cq}} \omega_{cq}^{(j)} h(\boldsymbol{\chi}'_j)$, then update the filtered mean and its corresponding error covariance by

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k) \\
\boldsymbol{K}_k &= (\boldsymbol{P}_{xy}/\boldsymbol{S}_{yy}^{\mathrm{T}})/\boldsymbol{S}_{yy} \\
\boldsymbol{S}_{k|k} &= uptria([\boldsymbol{X}'_{k|k-1} - \boldsymbol{K}_k\boldsymbol{Y}_{k|k-1}, \boldsymbol{K}_k\boldsymbol{S}_{R_k}]) \qquad (9)
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{Y}_{k|k-1} &= [h(\boldsymbol{\chi}'_1) - \hat{\boldsymbol{y}}_{k|k-1}, \ldots, h(\boldsymbol{\chi}'_{n_{cq}}) - \hat{\boldsymbol{y}}_{k|k-1}]\boldsymbol{\Pi} \\
\boldsymbol{X}'_{k|k-1} &= [\boldsymbol{\chi}'_1 - \hat{\boldsymbol{x}}_{k|k-1}, \ldots, \boldsymbol{\chi}'_{n_{cq}} - \hat{\boldsymbol{x}}_{k|k-1}]\boldsymbol{\Pi} \\
\boldsymbol{S}_{yy} &= uptria([\boldsymbol{Y}_{k|k-1}, \boldsymbol{S}_{R_k}]) \\
\boldsymbol{P}_{xy} &= \boldsymbol{X}'_{k|k-1}\boldsymbol{Y}_{k|k-1}^{\mathrm{T}}.
\end{aligned}$$

### 3.2 Iterated GCQKF

Inspired by the iterated EKF and the iterated UKF, we modify the measurement update of GCQKF to make the latest measurement fully exploited. We introduce the maximum likelihood based terminal criterion which can guarantee the increase of likelihood [24]. In real applications, the number of iteration $N$ is not too large, so the I-GCQKF almost has the same computational complexity as GCQKF does. Since I-GCQKF has the same initial set up and prediction step except determining a maximal iteration number $N$, we here focus on the measurement update step.

**Measurement update step** Set $\hat{\boldsymbol{x}}_{k|k}^{(0)} = \hat{\boldsymbol{x}}_{k|k-1}$, $\boldsymbol{P}_{k|k}^{(0)} = \boldsymbol{P}_{k|k-1}$ and then go to the iteration loop from $i = 1$ to $N$. Compute the Cholesky decomposition $\boldsymbol{P}_{k|k}^{(i-1)} = \boldsymbol{S}_{k|k}\boldsymbol{S}_{k|k}^{\mathrm{T}}$ and evaluate CQ points as $\boldsymbol{\chi}'_j = \boldsymbol{S}_{k|k}\boldsymbol{\xi}_j + \hat{\boldsymbol{x}}_{k|k}^{(i-1)}$. Update the filtering density at the $i$th iteration $p_{k|k} = \mathcal{N}(\hat{\boldsymbol{x}}_{k|k}^{(i)}, \boldsymbol{P}_{x|x}^{(i)})$ by

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k|k}^{(i)} &= \hat{\boldsymbol{x}}_{k|k}^{(i-1)} + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^{(i)}) \\
\boldsymbol{P}_{k|k}^{(i)} &= \boldsymbol{P}_{k|k}^{(i-1)} - \boldsymbol{K}_k\boldsymbol{P}_{yy}\boldsymbol{K}_k^{\mathrm{T}} \qquad (10)
\end{aligned}$$

where

$$\hat{\boldsymbol{y}}_k^{(i)} = \sum_{j=1}^{n_{cq}} \omega_{cq}^{(j)} h(\boldsymbol{\chi}'_j)$$

$$\boldsymbol{P}_{yy} = \sum_{j=1}^{n_{cq}} \omega_{cq}^{(j)} (h(\boldsymbol{\chi}'_j) - \hat{\boldsymbol{y}}_k^{(i)})(h(\boldsymbol{\chi}'_j) - \hat{\boldsymbol{y}}_k^{(i)})^{\mathrm{T}} + \boldsymbol{R}_k$$

$$\boldsymbol{P}_{xy} = \sum_{j=1}^{n_{cq}} \omega_{cq}^{(j)} (\boldsymbol{\chi}'_j - \hat{\boldsymbol{x}}_{k|k}^{(i-1)})(h(\boldsymbol{\chi}'_j) - \hat{\boldsymbol{y}}_k^{(i)})^{\mathrm{T}}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{xy}\boldsymbol{P}_{yy}^{-1}.$$

Calculate the terminal condition as

$$\begin{aligned}
&\boldsymbol{e}_x^{(j)\mathrm{T}}(\boldsymbol{P}_{k|k}^{(j-1)})^{-1}\boldsymbol{e}_x^{(j)} + \boldsymbol{e}_y^{(j)\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{e}_y^{(j)} \leqslant \\
&\qquad \boldsymbol{e}_y^{(j-1)\mathrm{T}}\boldsymbol{R}_k^{-1}\boldsymbol{e}_y^{(j-1)} \qquad (11)
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{e}_x^{(j)} &= \hat{\boldsymbol{x}}_{k|k}^{(j)} - \hat{\boldsymbol{x}}_{k|k}^{(j-1)} \\
\boldsymbol{e}_y^{(j)} &= \boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^{(j)}.
\end{aligned}$$

If the terminal condition holds or $j$ exceeds the maximal iteration number $N$, stop the iteration and set $\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k}^{(j)}$ and $\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k}^{(j)}$.

Akin to SR-GCQKF, the QR decomposition can be utilized in I-GCQKF to improve the numerical stability and estimation accuracy. It is easy to derive and thus is not provided here due to the space limitations.

## 4. Numerical simulations and discussions

In this section, the proposed algorithms have been applied to solve two typical nonlinear estimation problems. For convenience, in our notation, $\mathrm{C}_x\mathrm{Q}_y\mathrm{KF}$ is short for GCQKF in which the $x$th spherical cubature rule and the $y$th Gauss-Laguerre quadrature rule are applied. Both I-$\mathrm{C}_x\mathrm{Q}_y\mathrm{KF}$ and SR-$\mathrm{C}_x\mathrm{Q}_y\mathrm{KF}$ have the same meanings.

### 4.1 Single dimensional process

First introduced in [14], the single dimensional process has been widely used to test the tracking stability of the Gaussian approximation filters [18,19]. The plant model is given as

$$\begin{aligned}
x_k &= x_{k-1} + 5\Delta t x_{k-1}(1 - x_{k-1}^2) + \nu_{k-1} \\
y_k &= \Delta t x_k(1 - 0.5x_k) + \omega_k \qquad (12)
\end{aligned}$$

where $\nu_{k-1} \sim \mathcal{N}(0,\ \alpha^2\Delta t)$ and $\omega_k \sim \mathcal{N}(0,\ \beta^2\Delta t)$ are the uncorrelated white Gaussian noise; $\Delta t$ is the sample time. A time span from 0 to 2 s with the sample time $\Delta t = 0.1$ s has been considered in our study. The initial state and system parameters are set as: $x_0 = -0.4$, $\hat{x}_{0|0} = -0.8$, $P_{0|0} = 3$, $\alpha = 0.6$ and $\beta = 0.2$. This system has three equilibrium points, the first one at the origin and the other two at $\pm 1$. The first one is unstable while the other two are stable. The estimation error may force the state estimation to converge to the wrong equilibrium points, resulting in loss of tracking.

For comparing the stability performance of the delivered algorithms, we define a parameter called the percentage fail count, which means the ratio between the number of loss tracking and the number of Monte Carlo runs. The estimate error at the 2nd s is exploited as a criterion to determine the loss tracking or not. If the absolute estimate error at the 2nd s is more than one, we treat the filter as loss of tracking; otherwise the filter does not. In this paper, we execute 500 times of Monte Carlo runs. The percentage fail counts for different filters are showed in Table 1. From the table, we can see that square root algorithms are more stable than others and high order filters are more stable than low order ones. The stability performance of iterated algorithms is almost the same as that of the standard ones.

**Table 1 Percentage fail counts of different filters**

| Filter | Fail count/% | Filter | Fail count/% |
|---|---|---|---|
| $\mathrm{C_3Q_1KF}$ | 7.8 | I-$\mathrm{C_3Q_1KF}$ | 8 |
| | | SR-$\mathrm{C_3Q_1KF}$ | 3.8 |
| $\mathrm{C_5Q_1KF}$ | 3 | I-$\mathrm{C_5Q_1KF}$ | 3.4 |
| | | SR-$\mathrm{C_5Q_1KF}$ | 1.2 |
| $\mathrm{C_5Q_5KF}$ | 2.8 | I-$\mathrm{C_5Q_5KF}$ | 3 |
| | | SR-$\mathrm{C_5Q_5KF}$ | 1 |

### 4.2 Maneuvering target tracking

In this scenario, the maneuvering target tracking problem has been used to validate the performance of the developed algorithms. This is a benchmark problem which is widely used to test the performance of the Gaussian approximation filters [14,16–18]. The maneuvering target with unknown turn rate is modeled as

$$\boldsymbol{x}_{k+1}=\begin{pmatrix}1 & \dfrac{a_k}{\omega_k} & 0 & \dfrac{b_k-1}{\omega_k} & 0\\ 0 & b_k & 0 & -a_k & 0\\ 0 & \dfrac{1-b_k}{\omega_k} & 1 & \dfrac{a_k}{\omega_k} & 0\\ 0 & a_k & 0 & b_k & 0\\ 0 & 0 & 0 & 0 & 1\end{pmatrix}\boldsymbol{x}_k+\boldsymbol{\nu}_k \quad (13)$$

where $\boldsymbol{x}_k=[\alpha_k,\alpha'_k,\beta_k,\beta'_k,\omega_k]^{\mathrm{T}}$ is the state vector; $a_k=\sin(\omega_k\Delta t)$ and $b_k=\cos(\omega_k\Delta t)$ are two turn rate related terms; $[\alpha_k,\beta_k]$ are the position at time $k$ in the $x$ and $y$ directions respectively; $[\alpha'_k,\beta'_k]$ denotes the corresponding velocities and $\omega_k$ is the turn rate which is unknown; $\Delta t$ is the sample time; the process noise $\boldsymbol{\mu}_k$ is the white Gaussian noise with covariance $\boldsymbol{Q}_k=\mathrm{diag}[q_1\boldsymbol{M},q_1\boldsymbol{M},q_2\Delta t]$, where $\boldsymbol{M}$ is defined as

$$\boldsymbol{M}=\begin{pmatrix}\Delta t^3/3 & \Delta t^2/2\\ \Delta t^2/2 & \Delta t\end{pmatrix}.$$

The measurement is given by an active radar fixed at the origin as

$$\begin{pmatrix}r_{k+1}\\ \theta_{k+1}\end{pmatrix}=\begin{pmatrix}\sqrt{\alpha_{k+1}^2+\beta_{k+1}^2}\\ \arctan(\beta_{k+1}/\alpha_{k+1})\end{pmatrix}+\pi_{k+1} \quad (14)$$

where $\pi_{k+1}$ is the white Gaussian measurement noise with covariance $\boldsymbol{R}_{k+1}=\mathrm{diag}[\sigma_r^2,\sigma_\theta^2]$ and it is also assumed to be independent with the process noise $\mu_k$.

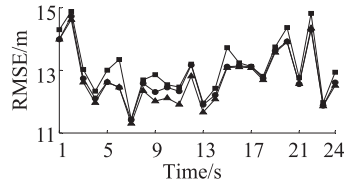
In the simulation, we use the following data: $\Delta t=1$ s, $\omega_0=-3^\circ\ \mathrm{s}^{-1}$, $q_1=0.1\ \mathrm{m}^2\cdot\mathrm{s}^{-3}$, $q_2=1.75\times10^{-4}\ \mathrm{s}^{-3}$, $\sigma_r=10$ m and $\sigma_\theta=\sqrt{10}$ mrad. To illustrate the effectiveness of the delivered algorithms, the initial estimate $\hat{\boldsymbol{x}}_{0|0}$ is randomly selected from $\mathcal{N}(\hat{\boldsymbol{x}}_{0|0};\boldsymbol{x}_{0|0},\boldsymbol{P}_{0|0})$, where $\boldsymbol{x}_{0|0}=[1\ 000,300,1\ 000,0,-3]$ and $\boldsymbol{P}_{0|0}=\mathrm{diag}[100,10,100,10,0.1]$. The simulations are executed 500 independent Monte Carlo runs in MATLAB in a computer with AMD Dual-core 2.30 GHz processor under the Windows 7 system.

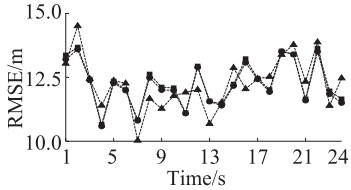
We introduce the root mean square error (RMSE) as a metric to evaluate the filtering performance and RMSE in position is defined as

$$\mathrm{RMSE}_{\mathrm{pos}}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}((\alpha_k^{(i)}-\hat{\alpha}_k^{(i)})^2+(\beta_k^{(i)}-\hat{\beta}_k^{(i)})^2)}$$

where $N$ is the number of Monto Carlo runs and $[\alpha_k^{(i)},\beta_k^{(i)}]$ and $[\hat{\alpha}_k^{(i)},\hat{\beta}_k^{(i)}]$ denote the true and estimated values of positions at $k$ instant in the $i$th Monte Carlo run respectively. Similarly the RMSE in velocity and turn rate also can be easily formulated.

The comparison of RMSE for different filters in position and velocity can be founded in Fig. 1 and Fig. 2 respectively. As can be seen, both I-GCQKF and SR-GCQKF outperform the same order GCQKF. It also can be illustrated that the higher order filters perform better than the lower order ones. However, no significant improvement is noticed with much higher orders. The RMSE in turn rate has the same trend. However, the results of distinct filters are quite similar due to no information in measurement, hence we do not show the RMSE in turn rate here.



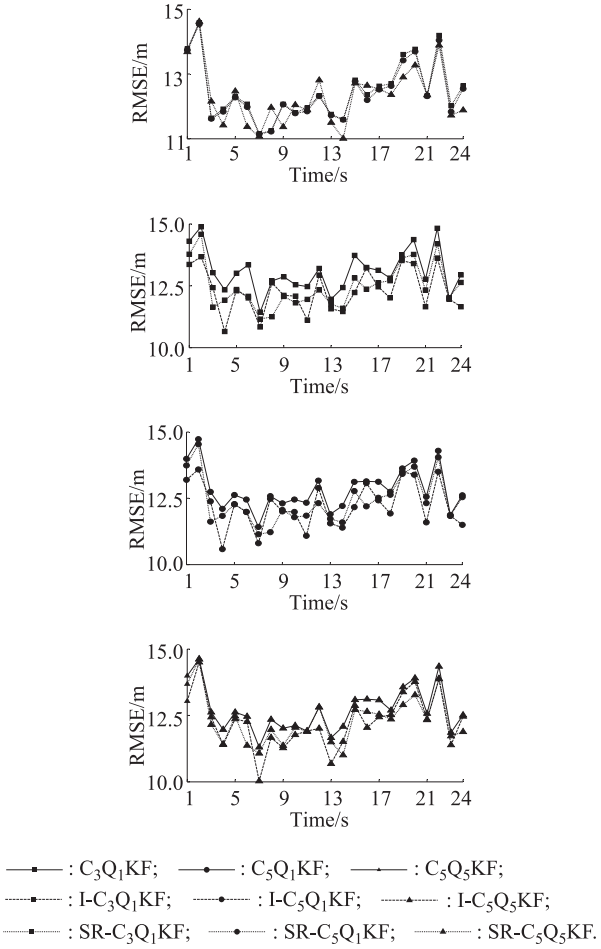
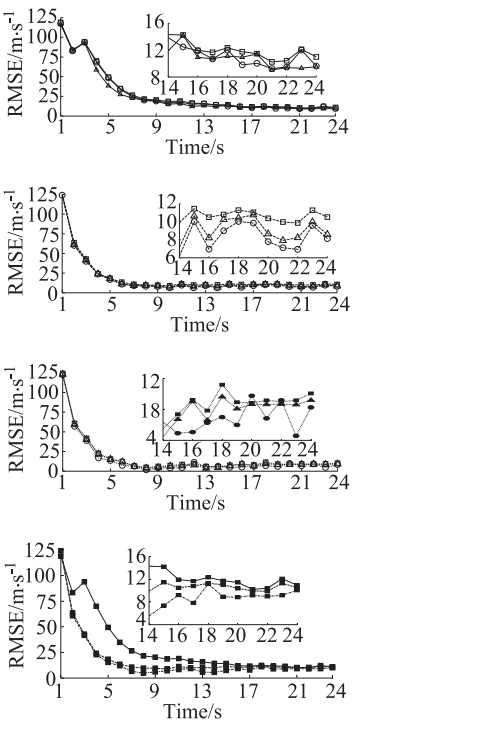



**Fig. 1 Comparisons of RMSE for different filters in position**



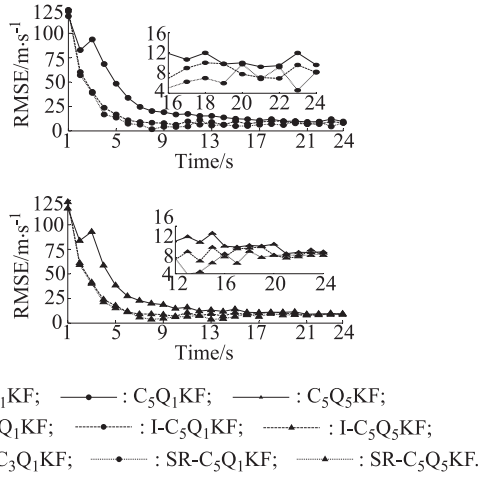




**Fig. 2 Comparisons of RMSE for different filters in velocity**

Computational time is also compared. The average execution time of the filters is showed in Table 2. As expected, higher order GCQKFs take more time than lower ones because more cubature quadrature points are possessed in higher order filters. When with the same order, square root filters spend almost the same but slightly little time as standard ones, while iterated filters need more time.

**Table 2 Average time consuming in different filters**

| Filter | Time/s | Filter | Time/s |
|---|---|---|---|
| $C_3Q_1KF$ | 0.032 7 | I-$C_3Q_1KF$ | 0.096 6 |
| | | SR-$C_3Q_1KF$ | 0.028 6 |
| $C_5Q_1KF$ | 0.126 9 | I-$C_5Q_1KF$ | 0.308 6 |
| | | SR-$C_5Q_1KF$ | 0.120 5 |
| $C_5Q_5KF$ | 0.4925 | I-$C_5Q_5KF$ | 0.985 6 |
| | | SR-$C_5Q_5KF$ | 0.490 5 |

## 5. Conclusions

In this paper, the derivation of GCQKF is introduced and the square root and iterated extensions of GCQKF are proposed. In lieu of applying the Cholesky decomposition, the QR decomposition is adopted in square root extensions at each iteration step, making the square root extensions more stable especially when facing the short computational length. In iterated extensions, the measurements are used several times to totally use the measurement information. Also a new terminal condition which guarantees the increase of likelihood functions has been applied to end the iterations.

Two numerical experiments are used to evaluate the performance of the proposed filters. The numerical results illustrate the superior performance of both iterated and square root extensions. However, whether iterated extensions outperform square root ones is problem-dependent, which needs further study. With the increase of the degree of cubature rule and order of quadrature rule, the number of the sigma points grow drastically, hence some technologies

such as marginalization which may decrease the numerical burden should be further studied.

## References


[1] S. Y. Cho, B. D. Kim. Adaptive IIR/FIR fusion filter and its application to the INS/GPS integrated system. *Automatica*, 2008, 44(8): 2040 – 2047.

[2] W. E. Leven, A. D. Lanterman. Unscented Kalman filters for multiple target tracking with symmetric measurement equations. *IEEE Trans. on Automatic Control*, 2009, 54(2): 370 – 375.

[3] S. Chatterjee, S. Sadhu, T. K. Ghoshal. Fault detection and identification of non-linear hybrid system using self-switched sigma point filter bank. *IET Control Theory & Applications*, 2014, 9(7): 1093 – 1102.

[4] G. A. Hossein, Z. X. Miao, L. L. Fan, et al. Identification of synchronous generator model with frequency control using unscented Kalman filter. *Electric Power Systems Research*, 2015, 126(2): 45 – 55.

[5] R. Jesse, M. Joaqułn. A distributed particle filter for nonlinear tracking in wireless sensor networks. *Signal Processing*, 2014, 98: 121 – 134.

[6] R. E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 1960, 82: 34 – 45.

[7] K. Ito, K. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Trans. on Automatic Control*, 2000, 45(5): 910 – 927.

[8] H. J. Kushner. Approximations to optimal nonlinear filters. *IEEE Trans. on Automatic Control*, 1967, 5(12): 546 – 556.

[9] J. Carpenter, P. Clifford, P. Fearnhead. Improved particle filter for nonlinear problems. *IEE Radar, Sonar and Navigation*, 1999, 146(1): 2 – 7.

[10] M. Simandl, J. Karlovec, T. Soderstorm. Advanced point-mass method for nonlinear state estimation. *Automatica*, 2006, 42(7): 1133 – 1145.

[11] O. Brun, V. Teuliere, J. M. Garcia. Parallel particle filtering. *Journal of Parallel and Distributed Computing*, 2002, 62(7): 1186 – 1202.

[12] H. Leung, Z. Zhu, Z. Ding. An aperiodic phenomenon of extended kalman filter in filtering noisy chaotic signals. *IEEE Trans. on Signal Processing*, 2000, 48(6): 1807 – 1810.

[13] S. Julier, J. Uhlmann, D. Whyte. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. on Automatic Control*, 2000, 45(3): 477 – 482.

[14] K. Ito, K. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Trans. on Automatic Control*, 2000, 45(5): 910 – 927.

[15] B. Jia, M. Xin, Y. Cheng. Sparse-grid Gaussian-Hermite quadrature nonlinear filtering. *Automatica*, 2012, 48(2): 327 – 341.

[16] I. Arasaratnam, S. Haykin. Cubature Kalman filters. *IEEE Trans. on Automatic Control*, 2009, 54(6): 1254 – 1269.

[17] B. Jia, M. Xin, Y. Cheng. High-degree cubature Kalman filter. *Automatica*, 2013, 49(2): 510 – 518.

[18] S. Bhaumik. Cubature quadrature Kalman filter. *IET Signal Processing*, 2012, 7(7): 533 – 541.

[19] K. S. Abhinoy, S. Bhaumik. Higher degree cubature quadrature Kalman filter. *International Journal of Control, Automation, and Systems*, 2015, 13(5): 1097 – 1105.

[20] S. B. Stoyanova. Cubature formulae of the seventh degree of accuracy for the hypersphere. *Journal of Computational and Applied Mathematics*, 1997, 84(1): 15 – 21.

[21] A. Genz. Fully symmetric interpolatory rules for multiple integrals over hyper spherical surface. *Journal of Computational and Applied Mathematics*, 2003, 157(1): 187 – 195.

[22] P. Silvester. Symmetric quadrature formulae for simplexes. *Mathematics of Computation*, 1970, 109(24): 95 – 100.

[23] I. Arasaratnam, S. Haykin. Square-root quadrature Kalman filtering. *IEEE Trans. on Signal Processing*, 2008, 56(6): 2589 – 2593.

[24] C. Wang, J. Zhang, J. Mu. Maximum likelihood-based iterated divided difference filter for nonlinear systems from discrete noisy measurements. *Sensors*, 2012, 12(7): 8912 – 8929.


## Biographies


**Hongwei Wang** was born in 1991. He received his B.S. degree in aerospace engineering from Northwestern Polytechnical University. He is now a Ph.D. candidate of Northwestern Polytechnical University. His research interests include statistical signal processing, nonlinear filtering, optimal estimation and data fusion.
E-mail: tianhangxinxiang@163.com



**Wei Zhang** was born in 1963. He received his B.S and M.S degrees in flight mechanics and Ph.D. degree in aircraft design from Northwestern Polytechnical University. He was invited as an associate professor in School of Creative Media, City University of Hong Kong during 2013 – 2014. Now he is a professor of School of Aeronautics, Northwestern Polytechnical University. His research interests include aircraft design methodology, system integrated and design, scaled flight testing and aircraft system identification techniques.
E-mail: weizhangxian@nwpu.edu.cn



**Junyi Zuo** was born in 1975. He received his M.S degree in flight mechanics and Ph.D. degree in control and information from Northwestern Polytechnical University. His research interests include flight mechanics and control system design, nonlinear filtering, information processing and aircraft parameter identification techniques.
E-mail: zuojunyi@163.com



**Heping Wang** was born in 1957. He received his Ph.D. degree from School of Aeronautics, Northwestern Polytechnical University in 1995. Currently, he is a professor at School of Aeronautics, Northwestern Polytechnical University. He is a member of Chinese Society of Aeronautics and Astronautics. His research interests include aircraft design and optimization design.
E-mail: sdmk163@163.com