

Article

Improved Cubature Kalman Filtering on Matrix Lie Groups Based on Intrinsic Numerical Integration Error Calibration with Application to Attitude Estimation

Huijuan Guo ^{*}, Yan Zhou, Huiying Liu and Xiaoxiang Hu

School of Automation, Northwestern Polytechnical University, Xi'an 710129, China; lea@mail.nwpu.edu.cn (Y.Z.); lhy2005@nwpu.edu.cn (H.L.); xxhu@nwpu.edu.cn (X.H.)

^{*} Correspondence: guohuijuan@mail.nwpu.edu.cn

Abstract: This paper investigates the numerical integration error calibration problem in Lie group sigma point filters to obtain more accurate estimation results. On the basis of the theoretical framework of the Bayes–Sard quadrature transformation, we first established a Bayesian estimator on matrix Lie groups for system measurements in Euclidean spaces or Lie groups. The estimator was then employed to develop a generalized Bayes–Sard cubature Kalman filter on matrix Lie groups that considers additional uncertainties brought by integration errors and contains two variants. We also built on the maximum likelihood principle, and an adaptive version of the proposed filter was derived for better algorithm flexibility and more precise filtering results. The proposed filters were applied to the quaternion attitude estimation problem. Monte Carlo numerical simulations supported that the proposed filters achieved better estimation quality than that of other Lie group filters in the mentioned studies.



Citation: Guo, H.; Zhou, Y.; Liu, H.; Hu, X. Improved Cubature Kalman Filtering on Matrix Lie Groups Based on Intrinsic Numerical Integration Error Calibration with Application to Attitude Estimation. *Machines* **2022**, *10*, 265. <https://doi.org/10.3390/machines10040265>

Academic Editors: Peter Odry, Akos Odry and Jan Awrejcewicz

Received: 1 March 2022

Accepted: 5 April 2022

Published: 7 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: Bayes–Sard quadrature moment transformation; cubature Kalman filter; Lie groups; attitude estimation

1. Introduction

Estimating the state of dynamical systems is a research hotspot and a frequently faced essential issue in many real-life applications, e.g., navigation, target tracking, robotics, and automatic control [1–3]. Due to the imperfect system model and the existence of noise, filtering is the most used estimating approach. For linear systems with Gaussian uncertainties, the Kalman filter (KF) [4], as the most desirable estimator established on minimal mean-square-error (MMSE) criterion, can provide an optimal recursive solution. However, while the system is nonlinear, obtaining the optimal closed-form solution is intractable. Researchers proposed various methods via analytical or numerical approximation to address nonlinear state estimation issues. A successful analytical approximation method is the extended Kalman filter (EKF) [5], which employs first-order Taylor series approximation to linearize nonlinear dynamics, but may perform poorly with highly nonlinear systems or large initial errors. Moreover, the main drawback of the EKF is that it needs to calculate the Jacobians, which is relatively tricky for complicated models.

Instead of dealing with nonlinearities by evaluating Jacobians, another class of nonlinear filtering based on numerical approximation is preferred. These filters are collectively called sigma point filters, including the unscented Kalman (UKF) [6], Gauss–Hermite–Kalman (GHKF) [7], and cubature Kalman (CKF) [8] filters. Different deterministic sigma points are used in these filtering algorithms to approximate moments of the probability distribution functions (PDFs). Compared with the EKF, the sigma point Kalman filter can provide better robustness and estimation accuracy, but increases the computational burden. Another problem underlying the sigma point Kalman filter is that errors caused by the traditional numerical quadrature rules may inject bias into the estimated moments without

compensation. Concerning that ignoring the integral approximation error may lead to overconfident estimates, several studies employed Bayesian quadrature (BQ) and other quadrature approaches based on BQ theory (such as Gaussian process quadrature (GPQ) and Bayes–Sard quadrature (BSQ)) to improve the sigma point Kalman filter [9–11]. These approaches are successful because BQ, GPQ, and BSQ rules can quantify uncertainty in numerical integration.

Most EKF and sigma point Kalman filters are designed for system dynamical models in Euclidean spaces in the existing literature. Nevertheless, when the system state exists in a Riemannian manifold, considering the geometry of the manifold can have merits, for example, enlarging the convergence domain and increasing the convergence rate [12]. Many works are devoted to introducing Euclidean space filters to manifolds to obtain natural and good metrics. The authors in [13] generalized the unscented transform and the UKF to Riemannian manifolds, and produced a general optimization framework. By extending the main concepts of UKF for Riemannian state-space systems, the authors in [14] proposed a series of more widespread and consistent Riemannian UKFs (RiUKFs) that supplement the theory gap in UKF on manifolds. Another generalized method about UKF on manifolds was presented in [15], which further developed the authors' previous work of UKF on Lie groups in [16,17].

Implementing KFs on Lie groups is another widely studied line of designing filtering algorithms on manifolds, which was first proposed in [18,19]. Due to the systems' invariance properties, many studies proposed invariant KFs on Lie groups: the authors in [20–23] designed intrinsic continuous-discrete invariant EKFs (IEKFs) for continuous-time systems on Lie groups with discrete-time measurements in Euclidean spaces and applied them to attitude estimation and inertial navigation; in [24–26], the researchers proposed the invariant UKFs (IUKFs). To help practitioners fully understand IEKFs, Ref. [27] provided three typical examples, and [28] presented the right IEKF algorithm for simultaneous localization and mapping (SLAM). Moreover, the authors in [29–32] devised EKFs on Lie groups for systems with measurements on matrix Lie groups. Other designs proposed EKF extensions for Lie groups [12,33]. On the other hand, UKF on Lie groups (UKF-LG) was developed in [16,17] for SLAM and sensor fusion. The authors in [34] suggested an impressive UKF based on UKF theory on Lie groups for visual-inertial SLAM. Unlike UKF-LG, in [35], the authors proposed another type of UKF filter for matrix Lie groups. This filter takes the time propagation step on the Lie algebra, improving computation efficiency. In addition, the authors in [36] introduced an invariant CKF on Lie groups for the attitude estimation problem from vector measurements.

Even if the above works succeeded in extending the sigma point filters to the Lie group, these filters still do not consider integration errors caused by quadrature rules. Hence, this paper is concentrated on calibrating the numerical integration error in the sigma point filtering algorithm on matrix Lie groups to improve filtering estimation quality. Motivated by the previous study on employing the Bayes–Sard quadrature (BSQ) method in Euclidean sigma point filtering, we attempted to extend the BSQ method to filters on matrix Lie groups. According to the theoretical framework of the Bayes–Sard quadrature moment transformation (BSQMT) on BSQ in [11], we first give a version of BSQMT that selects cubature points as its prepoints, and then leverage the BSQMT to design our method. Lastly, the proposed method was verified by simulations in an attitude estimation system. An attitude estimation system usually consists of gyroscopes that provide angular velocity measurements and at least two vector sensors (such as accelerometers, magnetometers, star trackers, and sun sensors) that provide at least two nonparallel vector measurements. Our main contributions are as follows.

- (1) Giving a generalized system measurement model covering measurements in both Euclidean spaces and Lie groups, and developing a Bayesian estimator by utilizing BSQMT with the generalized measurement model for the state estimation problem on Lie groups.

- (2) Deriving improved cubature Kalman filtering on matrix Lie groups with BSQMT to calibrate numerical integration errors, and introducing a method with the maximum likelihood principle to calculate adaptive expected model variance.
- (3) Applying the proposed Lie group filtering to quaternion attitude estimation problems, and providing numerical simulations to validate the effectiveness of the proposed filtering.

The rest of this paper is constructed as follows. Section 2 reviews the preliminary knowledge of the Lie group. Section 3 outlines a designed Bayesian estimation method on Lie groups through leveraging the BSQMT for two different types of measurements. Section 4 gives detailed derivations of the proposed filtering algorithm on Lie groups. In Section 5, numerical simulations about quaternion attitude estimation demonstrate the effectiveness of the proposed filters. Lastly, in Section 6, we draw conclusions and present further work.

2. Mathematical Preliminary

This section reviews the theory and some basic properties of Lie groups used in this work. The following contents are chiefly based on [22,37] and partly on [15].

2.1. Introduction to Matrix Lie Groups

Matrix Lie group: Let $G \subset \mathbb{R}^{N \times N}$ be a matrix Lie group that is a subset of the invertible square matrix and holds properties including group identity, multiplication map, and inversion map: $\mathbf{I}_{N \times N} \in G; \forall \chi_1, \chi_2 \in G, \chi_1 \chi_2 \in G; \forall \chi \in G, \chi^{-1} \in G$; where $\mathbf{I}_{N \times N}$ is the identity matrix of $\mathbb{R}^{N \times N}$. The matrix Lie group is also characterized by a smooth manifold structure, so that both multiplication and inversion maps are smooth operations. Each point on G , i.e., $\chi \in G$, can attach to a tangent space at that point, denoted by $T_\chi G$. Tangent space $T_\chi G$, defined by the derivative of any curve $\gamma(t)$, where $\gamma(0) = \chi$ (see the left plot of Figure 1), is a subvector space of $\mathbb{R}^{N \times N}$ with equal dimensions to those of G . Among all tangent spaces of G , the tangent space taken at group identity matrix $\mathbf{I}_{N \times N}$ and denoted as $T_I G$ is called Lie algebra.

Lie algebra: Let \mathfrak{g} denote the Lie algebra associated with a d -dimension Lie group; there is $T_I G := \mathfrak{g} \in \mathbb{R}^{d \times d}$. As Lie groups and Lie algebras lack mathematical tools, one needs to identify Lie algebra \mathfrak{g} to Euclidean space \mathbb{R}^d for convenient calculation. There is a linear bijection between \mathfrak{g} and \mathbb{R}^d , i.e., $[\cdot]_G^\vee : \mathfrak{g} \rightarrow \mathbb{R}^d$. This linear bijection is invertible, and its inverse map is defined by $[\cdot]_G^\wedge : \mathbb{R}^d \rightarrow \mathfrak{g}$. For example, assume vector $\boldsymbol{\eta} \in \mathbb{R}^d$ and Lie algebra $\boldsymbol{\eta}^\wedge \in \mathfrak{g}$; then, we have $[\boldsymbol{\eta}^\wedge]_G^\wedge = \boldsymbol{\eta}^\wedge \in \mathfrak{g}$ and $[\boldsymbol{\eta}^\wedge]_G^\vee = \boldsymbol{\eta} \in \mathbb{R}^d$. The middle plot of Figure 1 gives a more intuitive illustration.

Matrix Lie exponential and logarithm maps: Matrix Lie exponential map $\exp_G : \mathbb{R}^d \rightarrow G$ and its inverse map $\log_G : G \rightarrow \mathbb{R}^d$ (the matrix Lie logarithm map) give two bijections between a neighborhood of $0 \in \mathbb{R}^d$ and a neighborhood of group identity matrix $\mathbf{I}_{d \times d} \in G$. We define the matrix Lie exponential and logarithm maps as $\exp_G(\cdot) := \exp_m([\cdot]_G^\wedge)$ and $\log_G(\cdot) := [\log_m(\cdot)]_G^\vee$, respectively, where \exp_m and \log_m are matrix exponential and logarithm operations. Here, matrix exponential and logarithm operations present the link between a Lie group and its unique corresponding Lie algebra: $\exp_m : \mathfrak{g} \rightarrow G$ and $\log_m : G \rightarrow \mathfrak{g}$. For vector $\boldsymbol{\eta} \in \mathbb{R}^d$, we have $\exp_G(\boldsymbol{\eta}) = \exp_m(\boldsymbol{\eta}^\wedge)$. All maps among matrix Lie group G , Lie algebra \mathfrak{g} , and vector space \mathbb{R}^d are illustrated in detail on the right plot of Figure 1.

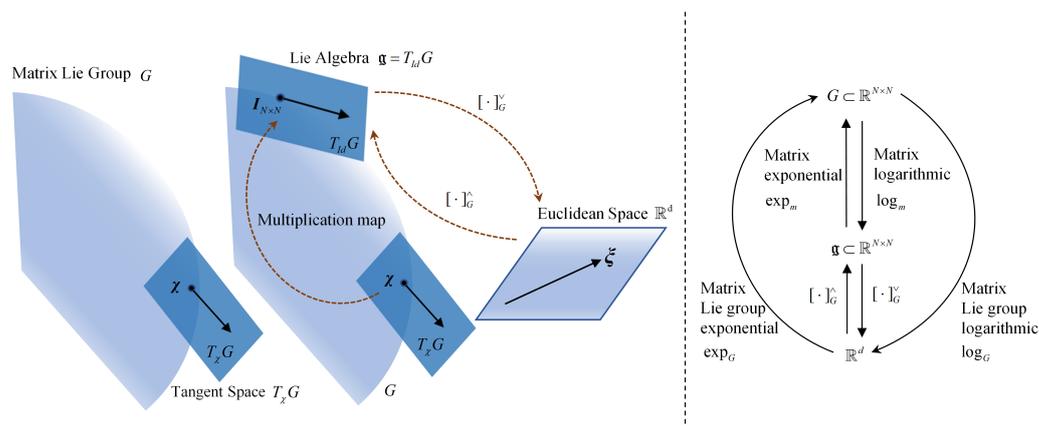


Figure 1. Schematic of mappings among Lie group, Lie algebra, and Euclidean space. Lie group model is oversimplified. Two distinct identifications of tangent space $T_\chi G$ at χ with tangent space $T_I G$ at $I_{N \times N}$ afforded by left or right Lie multiplication. Multiplication map includes left and right multiplication.

2.2. Uncertainty on Matrix Lie Groups

Most system noise is assumed to be additive white Gaussian noise in Euclidean spaces. However, the approach of additive Gaussian noise is not applicable to Lie groups, as Lie groups lack the addition operation. To describe Gaussian uncertainties for general Lie groups, we adopted the method in [15,16,34], which first assumes the distribution of uncertainty in the Lie algebra to be Gaussian and then maps Gaussian distribution to the corresponding Lie group by exponential mapping.

Consider a d -dimension random variable χ evolving on G , and let the variable satisfy prior probability distribution $\chi \sim \mathcal{N}_\phi(\hat{\chi}, P)$. Referring to the definition of the probability distribution for a random variable on a manifold in [15], we give the definition of probability distribution for $\chi \in G$ as follows:

$$\chi = \phi(\hat{\chi}, \eta), \eta \sim \mathcal{N}(0, P), \tag{1}$$

where $\phi : G \times \mathbb{R}^d \rightarrow G$ is a smooth function selected according to the composition of variable χ that must satisfy $\phi(\hat{\chi}, 0) = \hat{\chi}$, also named “retraction”; $\hat{\chi} \in G$ is noise-free and represents the mean of random variable χ ; operator $\mathcal{N}(\cdot, \cdot)$ denotes Gaussian distribution in a Euclidean space; $P \in \mathbb{R}^{d \times d}$ is the error covariance matrix associated with uncertainty perturbation η ; and $\eta \in \mathbb{R}^d$ is a random Gaussian vector in a Euclidean space. Moreover, distribution $\mathcal{N}_\phi(\cdot, \cdot)$ is not Gaussian.

The specific operation of function $\phi(\cdot, \cdot)$ is generally determined by the system’s geometric structure. Left and right Lie group multiplications, and the matrix Lie group exponential are essential parts of $\phi(\cdot, \cdot)$ no matter the system geometry. Here, with right and left multiplication, we give two simplified definitions of $\phi(\hat{\chi}, \eta)$, denoted by $\phi_R(\hat{\chi}, \eta) = \exp_G(\eta)\hat{\chi}$ and $\phi_L(\hat{\chi}, \eta) = \hat{\chi} \exp_G(\eta)$. The components of the variable evolving on G are different, so the expression of function $\phi(\cdot, \cdot)$ is different in various system dynamics and applications.

3. Bayesian Estimation Based on Bayes–Sard Quadrature Moment Transform

This section aims to find a method to calibrate numerical quadrature errors in the classical sigma point filtering on Lie groups. By taking the cubature Kalman filter into consideration and adopting the BSQMT instead of the cubature transform, we propose a Bayesian estimation method with a generalized measurement model.

3.1. Bayes–Sard Quadrature Moment Transform with Cubature Points

The BSQMT [11] is a universal moment transform proposed to accommodate improper estimation and prediction calibration caused by quadrature error in sigma point filters.

The main technique of the BSQMT is the Bayesian quadrature, which treats the numerical process as probabilistic inference, and models the integration error by utilizing stochastic process models. Consider nonlinear vector function

$$z = g(x), \tag{2}$$

where input $x \in \mathbb{R}^d$ is a Gaussian random variable that satisfies Gaussian distribution $\mathcal{N}(\bar{x}, P)$; $z \in \mathbb{R}^l$ is the output; $g(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^l$ denotes a known nonlinear transformation function. A brief generalization of the BSQMT algorithm with cubature points is given as follows:

Step 1. Assume that the mean \bar{x} and covariance P of random input variable x are known. Calculate third-degree cubature points that match the transform moment of random variable

$$X_i = \bar{x} + S\xi_i, i = 1, \dots, 2d, \tag{3}$$

where S represents the lower triangular factor in the Cholesky decomposition of P ; ξ_i denotes unit cubature points expressed as

$$\xi_i = \begin{cases} \sqrt{d}e_i, i = 1, \dots, d \\ -\sqrt{d}e_{i-d}, i = d + 1, \dots, 2d \end{cases} \tag{4}$$

where $e_i \in \mathbb{R}^d$ is the i -th column of identity matrix $I_{d \times d}$.

Step 2. Compute the quadrature weight: BSQMT uses a nonzero mean hierarchical Gaussian process (GP) prior model, and Bayes–Sard weights merely depend on the choice of sigma points and basic function space $\pi := \text{span}\{\psi_1, \dots, \psi_N\}$. Let $\psi(x) = x^\alpha, x \in \mathbb{R}^d$, where α is multi-index; then, $x^\alpha = x_1^{\alpha_1} \dots x_d^{\alpha_d}$ and $|\alpha| = |\alpha_1 + \dots + \alpha_d|$. Here, we consider a special base function space in which $N = 2d$ holds.

Lemma 1. Let $p(x) = \mathcal{N}(x | 0, I)$ be standard normal distribution, choose $2d$ cubature points by (4), and construct $2d$ dimensional function space

$$\pi = \{x_1, \dots, x_d, x_1^2, \dots, x_d^2\}. \tag{5}$$

Then, a set of quadrature weights $w = \{w_m, n = 1, \dots, 2d\}$ can decide the weighted mean of the state variable, and quadrature weights cohere with cubature transform weights $w_m = 1/2d$.

Proof of Lemma 1. See Appendix A. \square

With third-degree unit cubature points, quadrature weights are calculated with the following equations:

$$\begin{cases} w_m = \Psi^{-T}\bar{\psi} \\ w = \Psi^{-T}W_A\Psi^{-1} \\ w_c = W_B\Psi^{-1} \end{cases}, \tag{6}$$

in which

$$\begin{cases} \bar{\psi} = E_{\xi}[\psi(\xi)] \\ \Psi = [\psi(\xi_1), \dots, \psi(\xi_{2d})]^T \\ W_A = E_{\xi}[\psi(\xi)\psi(\xi)^T] \\ W_B = E_{\xi}[\xi\psi(\xi)^T] \end{cases}. \tag{7}$$

From Lemma 1, we know that $w_m = 1/2d$. Thus, we can skip the calculation of w_m in (6). Step 3. Obtain transformed moments with the following equations:

$$m_z = Z w_m, \tag{8}$$

$$P_{zz} = Z w Z^T - m_z m_z^T + \sigma^2 I, \tag{9}$$

$$C_{xz} = S w_c Z^T, \tag{10}$$

where $\mathbf{Z} = [g(\mathbf{X}_1), \dots, g(\mathbf{X}_{2d})]$. $\bar{\sigma}^2$ is expected model variance (EMV) of which the value partly depends on the selection of radial basis function (RBF) kernel (covariance of GP model based on unit cubature points) and its parameter. However, in practice, EMV sometimes needs to be reparameterized, but we omitted calculating it here. See [11,38] for the detailed computation of EMV and the selection rules of the basic function.

3.2. Bayesian Estimation on Lie Groups Using BSQMT

In this subsection, we first give a generalized measurement model covering system measurements in two kinds of spaces: Euclidean space and Lie group. Consider random variable $\chi \in G$ and its prior probability density function $p(\chi)$, χ follows (1) with known $\hat{\chi}$ and \mathbf{P} . Assume that one can establish additional information about the random variable via an observation \mathbf{y} :

$$\mathbf{y} = h_a(\chi, \mathbf{v}) = \begin{cases} h_{a1}(\chi) + \mathbf{v}, & \text{for } \mathbf{y} \in \mathbb{R}^l \\ \phi(\chi, \mathbf{v}), & \text{for } \mathbf{y} \in G \end{cases}, \quad (11)$$

where \mathbf{v} is white Gaussian noise with distribution $\mathbf{v} \sim \mathcal{N}(0, \mathbf{R})$ in the vector space, and its dimension depends on the type of \mathbf{y} ; $h_a(\cdot)$ and $h_{a1}(\cdot)$ represent the observation function, and $h_a(\cdot)$ satisfies the following mapping:

$$h_a(\cdot) : \begin{cases} G \rightarrow \mathbb{R}^l, & \text{for } \mathbf{y} \in \mathbb{R}^l \\ G \times \mathbb{R}^d \rightarrow G, & \text{for } \mathbf{y} \in G \end{cases} \quad (12)$$

The key to addressing the Bayesian estimation problem is to find parameters $\hat{\chi}^+$ and \mathbf{P}^+ in posterior distribution approximated by $p(\chi | \mathbf{y}) \approx \mathcal{N}_\phi(\hat{\chi}^+, \boldsymbol{\eta}^+)$, $\boldsymbol{\eta}^+ \sim \mathcal{N}(0, \mathbf{P}^+)$. Before estimating $\hat{\chi}^+$, posterior distribution $p(\boldsymbol{\eta} | \mathbf{y})$ should be found and approximated using BSQMT. Whether the measurement is in a vector space or a Lie group, there is an achievable way to calculate posterior estimate $\boldsymbol{\eta}^+$ in the vector space and then map it to the Lie group. Hence, while an available measurement \mathbf{y} exists in a Lie group, we need to transform the matrix Lie group measurement into the vector space by utilizing the inverse mapping of $\phi(\cdot, \cdot)$. Inverse map $\phi^{-1}(\cdot, \cdot) : G \rightarrow \mathbb{R}^d$ is defined as

$$\phi^{-1}(\hat{\chi}, \mathbf{y}) = \begin{cases} \phi_L^{-1}(\hat{\chi}, \mathbf{y}) = \log_G(\hat{\chi}^{-1}\mathbf{y}), & \text{left multiplication} \\ \phi_R^{-1}(\hat{\chi}, \mathbf{y}) = \log_G(\mathbf{y}\hat{\chi}^{-1}), & \text{right multiplication} \end{cases}. \quad (13)$$

In terms of left and right multiplications in both $\phi(\cdot, \cdot)$ and $\phi^{-1}(\cdot, \cdot)$, if \mathbf{y} exists in a Lie group, there are actually four possible expansions to map it to the vector space. For instance, similar to [16], suppose measurement function $\mathbf{y} = \chi \exp_G(\mathbf{v})$, if $\chi = \hat{\chi} \exp_G(\boldsymbol{\eta})$,

$$\begin{aligned} \mathbf{y}_L &= \phi_L^{-1}(\hat{\chi}, \mathbf{y}) = \log_G(\hat{\chi}^{-1}\chi \exp_G(\mathbf{v})) \\ &= \log_G(\hat{\chi}^{-1}\hat{\chi} \exp_G(\boldsymbol{\eta}) \exp_G(\mathbf{v})) \\ &= \log_G(\exp_G(\boldsymbol{\eta}) \exp_G(\mathbf{v})) \end{aligned}, \quad (14)$$

else if $\chi = \exp_G(\boldsymbol{\eta})\hat{\chi}$, there is

$$\begin{aligned} \mathbf{y}_L &= \phi_L^{-1}(\hat{\chi}, \mathbf{y}) = \log_G(\hat{\chi}^{-1}\chi \exp_G(\mathbf{v})) \\ &= \log_G(\hat{\chi}^{-1}\exp_G(\boldsymbol{\eta})\hat{\chi} \exp_G(\mathbf{v})) \\ &= \log_G(\exp_G(\text{Ad}_{\hat{\chi}^{-1}}(\boldsymbol{\eta})) \exp_G(\mathbf{v})) \end{aligned}, \quad (15)$$

where $\text{Ad}(\cdot)$ is the joint operator of a Lie group, there is $\text{Ad}_{\hat{\chi}^{-1}}(\boldsymbol{\eta}) = \hat{\chi}^{-1}\boldsymbol{\eta}\hat{\chi}$.

In addition, when an available measurement \mathbf{y} is in a vector space, it fulfils the following equation:

$$\mathbf{y} = h_{a1}(\boldsymbol{\chi}) + \mathbf{v} = h_{a1}(\phi(\hat{\boldsymbol{\chi}}, \boldsymbol{\eta})) + \mathbf{v}. \tag{16}$$

From (14)–(16), we can detect that the optimized measurement function is related to $\hat{\boldsymbol{\chi}}, \boldsymbol{\eta}$ and \mathbf{v} . So, giving a novel measurement equation for the derivation of the generalized nonlinear filtering:

$$\mathbf{y}' = h(\hat{\boldsymbol{\chi}}, \boldsymbol{\eta}, \mathbf{v}), \tag{17}$$

where \mathbf{y}' is the new observation, and $h(\cdot)$ is the new generalized measurement function.

By using BSQMT, propagated measurement cubature points and the measurement mean are computed:

$$\begin{cases} \hat{\mathbf{y}}'_i = h(\hat{\boldsymbol{\chi}}, \boldsymbol{\eta}_i, 0), i = 1, \dots, 2d \\ \bar{\mathbf{y}}' = \mathbf{Y}w_m \end{cases}, \tag{18}$$

where $\boldsymbol{\eta}_i$ is cubature points; $\mathbf{Y} = [\hat{\mathbf{y}}'_1, \dots, \hat{\mathbf{y}}'_{2d}]$, $i = 1, \dots, 2d$ and w_m is the mean weight defined in the above subsection. Afterwards, by calculating measurement error covariance \mathbf{P}_{yy} and cross-covariance $\mathbf{P}_{\eta y}$, we obtain gain matrix \mathbf{K} and parameters in the approximation of posterior distribution $p(\boldsymbol{\eta} | \mathbf{y}) \sim \mathcal{N}(\hat{\boldsymbol{\eta}}, \mathbf{P}^+)$:

$$\begin{cases} \mathbf{K} = \mathbf{P}_{\eta y} \mathbf{P}_{yy}^{-1} \\ \mathbf{P}^+ = \mathbf{P} - \mathbf{K} \mathbf{P}_{yy} \mathbf{K}^T \\ \hat{\boldsymbol{\eta}} = \mathbf{K}(\mathbf{y}' - \bar{\mathbf{y}}') \end{cases}. \tag{19}$$

For converting posterior distribution $p(\boldsymbol{\eta} | \mathbf{y})$ into a distribution on the Lie group, we reviewed analyses and results in [15,16] and obtained

$$\hat{\boldsymbol{\chi}}^+ = \phi(\hat{\boldsymbol{\chi}}, \hat{\boldsymbol{\eta}}) = \phi(\hat{\boldsymbol{\chi}}, \mathbf{K}(\mathbf{y}' - \bar{\mathbf{y}}')). \tag{20}$$

Lastly, the entire Bayesian estimation procedure is summarized in Algorithm 1.

Algorithm 1 Bayesian estimation on Lie groups using BSQMT.

Input: prior state $\hat{\boldsymbol{\chi}}$ and prior covariance \mathbf{P} , unit cubature points $\boldsymbol{\xi}_i, i = 1, 2, \dots, 2d$, sequence of observations \mathbf{y} , modified measurement noise covariance $\hat{\mathbf{R}}$, Bayes–Sard weights w_m, w, w_c , expected model variance $\bar{\sigma}_h^2$ (calculated by the method in [11])

Output: posterior state $\hat{\boldsymbol{\chi}}^+$ and associated posterior covariance \mathbf{P}^+

- 1: Calculate predicted measurement, and its associated cross-covariance and innovation covariance:

$$\begin{aligned} \boldsymbol{\eta}_i &= \text{chol}(\mathbf{P})\boldsymbol{\xi}_i, i = 1, \dots, 2d \\ \hat{\mathbf{y}}'_i &= h(\phi(\hat{\boldsymbol{\chi}}, \boldsymbol{\eta}_i, 0)), i = 1, \dots, 2d \\ \bar{\mathbf{y}}' &= \mathbf{Y}w_m, \text{ in which } \mathbf{Y} = [\hat{\mathbf{y}}'_1, \dots, \hat{\mathbf{y}}'_{2d}], i = 1, \dots, 2d \\ \mathbf{P}_{yy} &= \mathbf{Y}w\mathbf{Y}^T - \bar{\mathbf{y}}'\bar{\mathbf{y}}'^T + \bar{\sigma}_h^2\mathbf{I} + \hat{\mathbf{R}} \\ \mathbf{P}_{\eta y} &= \text{chol}(\mathbf{P})w_c\mathbf{Y}^T \end{aligned}$$

- 2: Calculate the gain matrix: $\mathbf{K} = \mathbf{P}_{\eta y} \mathbf{P}_{yy}^{-1}$;
- 3: Compute the estimated state and covariance:

$$\begin{aligned} \hat{\boldsymbol{\chi}}^+ &= \phi(\hat{\boldsymbol{\chi}}, \mathbf{K}(\mathbf{y}' - \bar{\mathbf{y}}')), \text{ where } \mathbf{y}' = \begin{cases} \mathbf{y}, & \text{for } \mathbf{y} \in \mathbb{R}^l \\ \phi^{-1}(\hat{\boldsymbol{\chi}}, \mathbf{y}), & \text{for } \mathbf{y} \in \mathbf{G} \end{cases} \\ \mathbf{P}^+ &= \mathbf{P} - \mathbf{K} \mathbf{P}_{yy} \mathbf{K}^T \end{aligned}$$

4. Proposed Filtering Algorithm

This section presents a Bayes–Sard cubature Kalman filter on matrix Lie groups built on numerical integration error calibration with BSQMT to correct the filtering accuracy and named BSCKF-LG. The proposed BSCKF-LG has two versions: one version uses left

multiplication to define “retraction” as left BSCKF-LG; the other uses right multiplication defined as right BSCKF-LG. A systemic method that leverages the maximum likelihood criterion is also proposed to estimate the EMVs. By reconstructing the covariances of the prior state error and the innovation with the innovation sequence in a sliding window, the proposed method estimates EMVs and feeds them back into improved filtering to adjust the Lie Kalman gain matrix.

4.1. Improved Cubature Kalman Filtering on Lie Groups with BSQMT

Consider discrete system dynamics

$$\begin{cases} \chi_k = f(\chi_{k-1}, \omega_k, \mathbf{n}_k) \\ \mathbf{y}_k = h_a(\chi_k, \mathbf{v}_k) \end{cases}, \quad (21)$$

where k represents the timestamp; χ_k is the system state living in a matrix Lie group; $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ and $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$ are white Gaussian noise with known covariances in the vector space; and ω_k is a known input. According to the above section, we first needed to remodel measurement \mathbf{y}_k into \mathbf{y}'_k , and \mathbf{y}'_k is in the Euclidean vector space.

The purpose of designing filtering in the Bayesian framework of this model is to approximate posterior distribution $p(\chi_k, \mathbf{P}_k | \mathbf{y}'_{1:k})$. The recursive solution comprises two steps to the proposed filtering problem on Lie groups: propagation and update.

Propagation: start with giving prior probability density in the Bayesian framework.

$$p(\chi_{k-1}) \approx \mathcal{N}_\phi(\hat{\chi}_{k-1}^+, \boldsymbol{\eta}_{k-1}^+), \boldsymbol{\eta}_{k-1}^+ \sim \mathcal{N}(0, \mathbf{P}_{k-1}^+), \quad (22)$$

where $\hat{\chi}_{k-1}^+$ and \mathbf{P}_{k-1}^+ are posterior estimation results of the previous moment. Then, try to approximate the state distribution of propagation:

$$p(\chi_k | \mathbf{y}'_{1:k-1}) \approx \mathcal{N}_\phi(\hat{\chi}_k, \boldsymbol{\eta}_k), \boldsymbol{\eta}_k \sim \mathcal{N}(0, \mathbf{P}_k). \quad (23)$$

Our work needs to find $\hat{\chi}_k$ and \mathbf{P}_k . In the above Bayesian estimation, posterior state mean $\hat{\chi}_k^+$ and its associated covariance \mathbf{P}_k^+ are estimated using the uncertainty represented by (1). Here, for Model (21), we still used this uncertainty representation and the associated inverse map that has

$$\phi^{-1}(\hat{\chi}_k, \chi_k) = \phi^{-1}(\hat{\chi}_k, \phi(\hat{\chi}_k, \boldsymbol{\eta}_k)) = \log(\exp(\boldsymbol{\eta}_k)) = \boldsymbol{\eta}_k + O(\|\boldsymbol{\eta}_k\|^2), \quad (24)$$

where $O(\|\boldsymbol{\eta}_k\|^2)$ stands for error term. The proposed filter first propagates state mean $\hat{\chi}_k$ via employing the system’s deterministic part (noiseless state model), that is,

$$\hat{\chi}_k = f(\hat{\chi}_{k-1}^+, \omega_k, 0). \quad (25)$$

Subsequently, to compute associated covariance \mathbf{P}_k , we selected a set of unit cubature points for the BSQMT method and generated sigma points $\boldsymbol{\eta}_{k-1,i}^+$ in vector space \mathbb{R}^d through \mathbf{P}_{k-1}^+ . Then, mapping these points to the matrix Lie group and passing the mapped cubature points through noise-free Model (25), we could obtain propagated sigma points χ_k^i . On the basis of this work and with defined inverse map $\phi^{-1}(\hat{\chi}_k, \chi_k^i)$, $i = 1, \dots, 2d$, we obtained prior covariance \mathbf{P}_k (see Algorithm 2 for details).

Update: in this procedure, we need to approximate posterior probability distribution.

$$p(\chi_k | \mathbf{y}'_{1:k}) \approx \mathcal{N}_\phi(\hat{\chi}_k^+, \boldsymbol{\eta}_k^+), \boldsymbol{\eta}_k^+ \sim \mathcal{N}(0, \mathbf{P}_k^+), \quad (26)$$

when each available measurement \mathbf{y}_k arrives. Due to the same measurement model, the Bayesian estimation method in Section 3 was applied to compute the posterior state and its associated covariance. The proposed filtering algorithm executes the propagation step and

does not perform the update step until the measurement arrives. Algorithm 2 summarizes the proposed filter in detail.

Algorithm 2 Bayes–Sard cubature Kalman filter (BSCKF-LG).

Input: initial conditions χ_0 and $P_{0|0}$, sequence of inputs $\omega_k, k = 1, 2, \dots$, modified process and measurement noise covariances \hat{Q}_k and \hat{R}_k , sequence of observations $y_k, k = 1, 2, \dots$, RBF kernel parameters θ_f and θ_h (for computing EMV), unit cubature points $\xi_i, i = 1, 2, \dots, 2d$

Output: posterior state $\hat{\chi}_k^+$, posterior covariance P_k^+

- 1: Initialization: $\hat{\chi}_0^+ = \chi_0$ and $P_0^+ = P_{0|0}$;
- 2: Compute Baye–Sard quadrature weights w_m, w , and w_c from (6) and (7), and calculate the EMV $\bar{\sigma}_f^2$ with the method in [11];
- 3: **for** $k = 1, 2, 3, \dots$ **do**
 Propagation:
- 4: Calculate state mean through noiseless system model

$$\hat{\chi}_k = f(\hat{\chi}_{k-1}^+, \omega_k, 0)$$

- 5: Generate propagated sigma points χ_k^i on the basis of selected unit cubature points according to (4),

$$\begin{aligned} \eta_{k-1,i}^+ &= \text{chol}(P_{k-1}^+) \xi_i, i = 1, \dots, 2d \\ \chi_k^i &= f(\phi(\hat{\chi}_{k-1}^+, \eta_{k-1,i}^+), \omega_k, 0), i = 1, \dots, 2d \end{aligned}$$

- 6: Calculate associated covariance according to inverse map $\phi^{-1}(\hat{\chi}_k, \chi_k^i), i = 1, \dots, 2d$,

$$P_k = \Phi_k w \Phi_k^T + \bar{\sigma}_f^2 I + \hat{Q}_k$$

where $\Phi_k = [\phi^{-1}(\hat{\chi}_k, \chi_k^1), \dots, \phi^{-1}(\hat{\chi}_k, \chi_k^{2d})]$, \hat{Q}_k derived from the dynamical equation of η_k ;

Update: (If the measurement y_k arrives, start the update step of the filtering. BSQ weights and EMV value are known.)

- 7: Compute $\hat{\chi}_k^+, P_k^+$ from Algorithm 1 with $\hat{\chi}_k, P_k$
 - 8: **end for**
-

Remark 1. The propagation of state means does not need to utilize the BSQMT (or other moment transformation) because (25) was validated by [16] up to the second order. Instead of generating the sigma points directly by the distribution on the Lie group, all work about generating the corresponding sigma points is first done in the Euclidean space and then mapped to the Lie group.

Remark 2. The unscented Kalman filters on Lie groups in [15–17] generate sigma points for the process noise and calculate the covariance matrix of the deviation due to noise. Unlike these filters, process and measurement noise covariances \hat{Q}_k and \hat{R}_k in Algorithm 2 are determined by the specified dynamic equation of η_k and the remodeled measurement function of y'_k to save computing resources.

Remark 3. BSQMT rules define all Bayes–Sard weights. Weights only depend on the selection of the sigma points and the basic function. The proposed filter requires two different kernel parameter values to obtain EMV values (see [11] for details) because there are two integrated functions. The kernel parameter’s misspecification affects the BSQMT little, so manual parameter tuning of the EMV is available in practice. In the following section, we give a method to obtain the time-varying value of EMV.

4.2. EMV Estimation

In this subsection, we attempt to develop an adaptive BSCKF-LG with time-varying EMVs. To derive the EMV at each instant k , we first assumed that the prior covariance and the innovation covariance are unknown constants. The innovation vector is denoted as $\tilde{\mathbf{y}}'_k = \mathbf{y}'_k - \hat{\mathbf{y}}'_k$, and we consider the following maximum likelihood function of the rebuild measurements $\{\mathbf{y}'_{k-n+1}, \mathbf{y}'_{k-n+2}, \dots, \mathbf{y}'_k\}$:

$$\begin{aligned} L(\mathbf{P}_{yy,k}(\text{or } \mathbf{P}_k) | \mathbf{y}'_{k-n+1:k}) &= \ln \prod_{j=k-n+1}^k p(\mathbf{y}'_j | \mathbf{P}_{yy,j}(\text{or } \mathbf{P}_j)) \\ &= \ln \prod_{j=k-n+1}^k \frac{1}{\sqrt{(2\pi)^l |\mathbf{P}_{yy,j}|}} \exp\left(-\frac{1}{2} \tilde{\mathbf{y}}_j^T \mathbf{P}_{yy,j}^{-1} \tilde{\mathbf{y}}_j\right) \quad , \quad (27) \\ &= -\frac{1}{2} \sum_{j=k-n+1}^k \left\{ \ln(|\mathbf{P}_{yy,j}|) + \tilde{\mathbf{y}}_j^T \mathbf{P}_{yy,j}^{-1} \tilde{\mathbf{y}}_j \right\} + C \end{aligned}$$

where $|\cdot|$ denotes the determinant operator, n represents the sliding window size, and C is the constant term. On the basis of (27), the maximum likelihood estimation of the error covariances of the prior state and the prior measurement is derived by

$$\hat{\mathbf{P}}_{yy,k|ML}(\hat{\mathbf{P}}_{k|ML}) = \arg \max L(\mathbf{P}_{yy,k}(\text{or } \mathbf{P}_k) | \mathbf{y}'_{k-n+1:k}). \quad (28)$$

To solve (28), we need to take the partial derivatives of $L(\mathbf{P}_{yy,k}(\text{or } \mathbf{P}_k) | \mathbf{y}'_{k-n+1:k})$ with respect to estimated prior covariance $\hat{\mathbf{P}}_k$ and estimated innovation covariance $\hat{\mathbf{P}}_{yy,k}$, and let them be equal to zero:

$$\frac{\partial L}{\partial \hat{\mathbf{P}}_{yy,k}} = 0 \quad \text{and} \quad \frac{\partial L}{\partial \hat{\mathbf{P}}_k} = 0. \quad (29)$$

Subsequently, the maximum likelihood equations are obtained via Equation (29):

$$\frac{\partial L}{\partial \hat{\mathbf{P}}_{yy,k}} = -\frac{1}{2} \text{tr} \left\{ \sum_{j=k-n+1}^k \left[\left(\mathbf{P}_{yy,j}^{-1} - \mathbf{P}_{yy,j}^{-1} \tilde{\mathbf{y}}_j \tilde{\mathbf{y}}_j^T \mathbf{P}_{yy,j}^{-1} \right) \frac{\partial \mathbf{P}_{yy,j}}{\partial \hat{\mathbf{P}}_{yy,k}^{a_1, b_1}} \right] \right\}, \quad (30)$$

and

$$\frac{\partial L}{\partial \hat{\mathbf{P}}_k} = -\frac{1}{2} \text{tr} \left\{ \sum_{j=k-n+1}^k \left[\left(\mathbf{P}_{yy,j}^{-1} - \mathbf{P}_{yy,j}^{-1} \tilde{\mathbf{y}}_j \tilde{\mathbf{y}}_j^T \mathbf{P}_{yy,j}^{-1} \right) \frac{\partial \mathbf{P}_{yy,j}}{\partial \hat{\mathbf{P}}_k^{a_2, b_2}} \right] \right\}, \quad (31)$$

where $\text{tr}(\cdot)$ denotes the trace operator; $\hat{\mathbf{P}}_{yy,k}^{a_1, b_1}$ corresponds to the a_1 th row, b_1 th column of $\hat{\mathbf{P}}_{yy,k}$ with $a_1, b_1 \in [1, \dots, l]$; similarly, $\hat{\mathbf{P}}_k^{a_2, b_2}$ corresponds to the a_2 th row, b_2 th column of $\hat{\mathbf{P}}_k$ with $a_2, b_2 \in [1, \dots, d]$.

Equations (29) and (30) show that

$$\text{tr} \left\{ \sum_{j=k-n+1}^k \left[\left(\mathbf{P}_{yy,j}^{-1} - \mathbf{P}_{yy,j}^{-1} \tilde{\mathbf{y}}_j \tilde{\mathbf{y}}_j^T \mathbf{P}_{yy,j}^{-1} \right) \frac{\partial \mathbf{P}_{yy,j}}{\partial \hat{\mathbf{P}}_{yy,k}^{a_1, b_1}} \right] \right\} = 0. \quad (32)$$

Therefore, the problem of estimating $\mathbf{P}_{yy,k}$ is actually computing the derivation of the innovation covariance with respect to $\mathbf{P}_{yy,j}, j = k-n+1, \dots, k$. As stated above, state errors are mapped from the Lie group into the Euclidean space, and prior state error covariance in the Euclidean space can be expressed as $\mathbf{P}_k = \mathbb{E}[\boldsymbol{\eta}_k \boldsymbol{\eta}_k^T]$. In this article, $\boldsymbol{\eta}_k = \phi^{-1}(\hat{\boldsymbol{\chi}}_k, \boldsymbol{\chi}_k)$ and $\boldsymbol{\eta}_k^+ = \phi^{-1}(\hat{\boldsymbol{\chi}}_k^+, \boldsymbol{\chi}_k)$. Using (21) and the formula of \mathbf{P}_k in Algorithm 2, we first have

$$\begin{aligned} \eta_k &= \phi^{-1}\left(f\left(\hat{\chi}_{k-1}^+, \omega_k\right), f\left(\chi_{k-1}, \omega_k, \mathbf{n}_k\right)\right) \\ &= \phi^{-1}\left(f\left(\hat{\chi}_{k-1}^+, \omega_k\right), f\left(\phi\left(\chi_{k-1}^+, \eta_{k-1}^+\right), \omega_k, \mathbf{n}_k\right)\right). \end{aligned} \tag{33}$$

Then, $f(\cdot)$ and (33) are expressed by the Taylor series expansion about η_{k-1}^+ , the estimation error η_k becomes

$$\eta_k = F_k \eta_{k-1}^+ + O\left(\left\|\eta_{k-1}^+\right\|\right) + \hat{\mathbf{n}}_k, \tag{34}$$

in which $F = d\phi^{-1}\left(f\left(\hat{\chi}_{k-1}^+, \omega_k\right), f\left(\phi\left(\chi_{k-1}^+, \eta_{k-1}^+\right), \omega_k\right)\right) / d\eta_{k-1}^+$ and $O\left(\left\|\eta_{k-1}^+\right\|\right)$ represent the second- and high-order moments about η_{k-1}^+ ; $\hat{\mathbf{n}}_k$ is derived from the specified dynamic equation of η_k , and its covariance is $\hat{\mathbf{Q}}_k$. To obtain exact equality, we consider the entire high order of η_{k-1}^+ by introducing a diagonal matrix $\alpha_k = \text{diag}(\alpha_{1,k}, \dots, \alpha_{d,k})$, which can scale the approximation error [39,40]:

$$\eta_k = \alpha_k F_k \eta_{k-1}^+ + \hat{\mathbf{n}}_k. \tag{35}$$

Similarly, by substituting (18) in $\tilde{\mathbf{y}}'_k = \mathbf{y}'_k - \bar{\mathbf{y}}'_k$ and expanding $h(\cdot)$ using the Taylor series about η_k , we have

$$\tilde{\mathbf{y}}'_k = \mathbf{H}_k \eta_k + O\left(\left\|\eta_k\right\|\right) + \hat{\mathbf{v}}_k, \tag{36}$$

where $\mathbf{H}_k = dh(\hat{\chi}_k, \eta_k) / d\eta_k$, $O\left(\left\|\eta_k\right\|\right)$ is the second- and higher-order moment of η_k ; $\hat{\mathbf{v}}_k$ is defined by the remodeled measurement formula, and its covariance is $\hat{\mathbf{R}}_k$. To consider the entire higher order, we also introduce a diagonal matrix $\beta_k = \text{diag}(\beta_{1,k}, \dots, \beta_{l,k})$ [39]:

$$\tilde{\mathbf{y}}'_k = \beta_k \mathbf{H}_k \eta_k + \hat{\mathbf{v}}_k, \tag{37}$$

Hence, prior state error covariance and innovation covariance can be expressed by

$$\mathbf{P}_k = \alpha_k F_k \mathbf{P}_{k-1}^+ F_k^T \alpha_k + \hat{\mathbf{Q}}_k, \tag{38}$$

$$\mathbf{P}_{yy,k} = \beta_k \mathbf{H}_k \mathbf{P}_k \mathbf{H}_k^T \beta_k + \hat{\mathbf{R}}_k. \tag{39}$$

Lastly, for linear system models (35) and (37), if the Kalman filtering is steady, \mathbf{P}_k converges to a constant matrix; \mathbf{P}_k^+ , \mathbf{K}_k and $\mathbf{P}_{yy,k}$ also become constant matrices [41].

On the basis of the above analysis, we assumed that our filtering process inside the sliding window was in a steady state. Consequently, the approximation of $\mathbf{P}_{yy,k}$ in the sliding window was around constant, and there exists $\mathbf{P}_{yy,j} = \mathbf{P}_{yy,k}$, $j = k - n + 1, \dots, k$ such that $\partial \mathbf{P}_{yy,j} / \partial \hat{\mathbf{P}}_{yy,k}^{a_1, b_1} = \partial \mathbf{P}_{yy,k} / \partial \hat{\mathbf{P}}_{yy,k}^{a_1, b_1}$ are 1 for the a_1 row and b_1 column, while other elements are zeros. In light of the trace definition, we simplified (32) to

$$\sum_{j=k-n+1}^k \left(\mathbf{P}_{yy,j}^{-1} - \mathbf{P}_{yy,j}^{-1} \tilde{\mathbf{y}}'_j \tilde{\mathbf{y}}_j^T \mathbf{P}_{yy,j}^{-1} \right) = 0. \tag{40}$$

Since $\mathbf{P}_{yy,j}$ was approximately constant in the window, we pre- and postmultiplied both sides of (40) with $\mathbf{P}_{yy,j}$:

$$\hat{\mathbf{P}}_{yy,k} = \frac{1}{n} \sum_{j=k-n+1}^k \hat{\mathbf{P}}_{yy,j} = \frac{1}{n} \sum_{j=k-n+1}^k \left(\tilde{\mathbf{y}}'_j \tilde{\mathbf{y}}_j^T \right). \tag{41}$$

Substituting $\mathbf{P}_{yy} = \mathbf{Y} \omega \mathbf{Y}^T - \bar{\mathbf{y}}' \bar{\mathbf{y}}'^T + \bar{\sigma}_h^2 \mathbf{I} + \hat{\mathbf{R}}$ in (41) and letting $\Delta_h = \bar{\sigma}_h^2 \mathbf{I}$, the EMV of the measurement function is described as

$$\hat{\Delta}_h = \hat{\sigma}_h^2 \mathbf{I} = \frac{1}{n} \sum_{j=k-n+1}^k \left[\tilde{\mathbf{y}}'_j \tilde{\mathbf{y}}_j^T - \mathbf{Y}_j \omega \mathbf{Y}_j^T + \bar{\mathbf{y}}'_j \bar{\mathbf{y}}_j^T - \hat{\mathbf{R}}_j \right]. \tag{42}$$

Next, similar steps were employed to estimate the EMV of the system function. Equations (29) and (31) show that

$$\begin{aligned} & \text{tr} \left\{ \sum_{j=k-n+1}^k \left[\left(\mathbf{P}_{yy,j}^{-1} - \mathbf{P}_{yy,j}^{-1} \tilde{\mathbf{y}}_j' \tilde{\mathbf{y}}_j^T \mathbf{P}_{yy,j}^{-1} \right) \frac{\partial \mathbf{P}_{yy,j}}{\partial \hat{\mathbf{P}}_k^{a_2, b_2}} \right] \right\} \\ &= \text{tr} \left\{ \sum_{j=k-n+1}^k \left[\left(\mathbf{P}_{yy,j}^{-1} - \mathbf{P}_{yy,j}^{-1} \tilde{\mathbf{y}}_j' \tilde{\mathbf{y}}_j^T \mathbf{P}_{yy,j}^{-1} \right) \mathbf{H}_k \frac{\partial \mathbf{P}_j}{\partial \hat{\mathbf{P}}_k^{a_2, b_2}} \mathbf{H}_k^T \right] \right\}, \tag{43} \\ &= 0 \end{aligned}$$

where $\mathbf{H}_k = \mathbf{P}_{\eta y, k}^T \mathbf{P}_k^{-1}$ or $\mathbf{H}_k = d\mathbf{y}'_k / d\eta_k$.

Again, since the filtering process was assumed to be stable in the sliding window, approximations of \mathbf{P}_k in the window were almost constant, $\mathbf{P}_j = \mathbf{P}_k, j = k - n + 1, \dots, k$, such that $\partial \mathbf{P}_j / \partial \hat{\mathbf{P}}_k^{a_2, b_2} = \partial \mathbf{P}_k / \partial \hat{\mathbf{P}}_k^{a_2, b_2}$ are 1 for the a_2 row and b_2 column, while other elements are zeros. Equation (43) was pre- and postmultiplied by \mathbf{H}^T and \mathbf{H}^{-T} ; then, in terms of the trace definition, we simplified (43) to

$$\sum_{j=k-n+1}^k \left[\mathbf{H}_j^T \left(\mathbf{P}_{yy,j}^{-1} - \mathbf{P}_{yy,j}^{-1} \tilde{\mathbf{y}}_j' \tilde{\mathbf{y}}_j^T \mathbf{P}_{yy,j}^{-1} \right) \mathbf{H}_j \right] = 0. \tag{44}$$

In sigma point filters, there exists $\mathbf{K}_k = \mathbf{P}_{\eta y, k} \mathbf{P}_{yy, k}^{-1} = \mathbf{P}_k \mathbf{H}_k^T \mathbf{P}_{yy, k}^{-1}$; thus, (44) can be transformed into

$$\begin{aligned} & \sum_{j=k-n+1}^k \left[\mathbf{P}_j^{-1} \left(\mathbf{K}_j \mathbf{H}_j \mathbf{P}_j - \mathbf{K}_j \tilde{\mathbf{y}}_j' \tilde{\mathbf{y}}_j^T \mathbf{K}_j^T \right) \mathbf{P}_j^{-1} \right] \\ &= \mathbf{P}_j^{-1} \sum_{j=k-n+1}^k \left[\left(\mathbf{K}_j \mathbf{H}_j \mathbf{P}_j - \mathbf{K}_j \tilde{\mathbf{y}}_j' \tilde{\mathbf{y}}_j^T \mathbf{K}_j^T \right) \right] \mathbf{P}_j^{-1}. \tag{45} \\ &= 0 \end{aligned}$$

(45) is satisfied if

$$\sum_{j=k-n+1}^k \left[\left(\mathbf{K}_j \mathbf{H}_j \mathbf{P}_j - \mathbf{K}_j \tilde{\mathbf{y}}_j' \tilde{\mathbf{y}}_j^T \mathbf{K}_j^T \right) \right] = 0. \tag{46}$$

From Algorithm 2, we know that

$$\begin{cases} \mathbf{K}_k \tilde{\mathbf{y}}_k' = \phi^{-1}(\hat{\boldsymbol{\lambda}}_k, \hat{\boldsymbol{\lambda}}_k^+) \\ \mathbf{P}_k - \mathbf{P}_k^+ = \mathbf{K}_k \mathbf{P}_{yy} \mathbf{K}_k^T = \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k \end{cases}. \tag{47}$$

Substituting (47) into (46), there readily is

$$\sum_{j=k-n+1}^k \left(\mathbf{P}_j - \mathbf{P}_j^+ - \phi^{-1}(\hat{\boldsymbol{\lambda}}_k, \hat{\boldsymbol{\lambda}}_k^+) \left(\phi^{-1}(\hat{\boldsymbol{\lambda}}_k, \hat{\boldsymbol{\lambda}}_k^+) \right)^T \right) = 0, \tag{48}$$

then

$$\sum_{j=k-n+1}^k \mathbf{P}_j = \sum_{j=k-n+1}^k \left(\mathbf{P}_j^+ + \phi^{-1}(\hat{\boldsymbol{\lambda}}_k, \hat{\boldsymbol{\lambda}}_k^+) \left(\phi^{-1}(\hat{\boldsymbol{\lambda}}_k, \hat{\boldsymbol{\lambda}}_k^+) \right)^T \right), \tag{49}$$

with $\mathbf{P}_j = \mathbf{P}_k, j < k$ in the window, the maximum likelihood of prior error covariance \mathbf{P}_k could be acquired from

$$\hat{\mathbf{P}}_k = \frac{1}{n} \sum_{j=k-n+1}^k \mathbf{P}_j = \frac{1}{n} \sum_{j=k-n+1}^k \left(\mathbf{P}_j^+ + \phi^{-1}(\hat{\boldsymbol{\lambda}}_k, \hat{\boldsymbol{\lambda}}_k^+) \left(\phi^{-1}(\hat{\boldsymbol{\lambda}}_k, \hat{\boldsymbol{\lambda}}_k^+) \right)^T \right). \tag{50}$$

Combining $P_k = \Phi w \Phi^T + \bar{\sigma}_f^2 I + \hat{Q}_k$ and (50), and letting $\Delta_f = \bar{\sigma}_f^2 I$, the estimated EMV could be described as

$$\hat{\Delta}_f = \hat{\sigma}_f^2 I = \frac{1}{n} \sum_{j=k-n+1}^k \begin{bmatrix} P_j^+ + \phi^{-1}(\hat{\chi}_k, \hat{\chi}_k^+) (\phi^{-1}(\hat{\chi}_k, \hat{\chi}_k^+))^T \\ -\Phi_j w \Phi_j^T - \hat{Q}_j \end{bmatrix}. \tag{51}$$

Remark 4. As EMVs must be non-negative ($\bar{\sigma}_f^2 \geq 0, \bar{\sigma}_h^2 \geq 0$) and Δ_f, Δ_h are diagonal matrices, so only estimating the diagonal elements of the EMV matrix is enough. Diagonal elements need to be non-negative; if some diagonal elements are negative, one should set them to zero.

Remark 5. Equations (42) and (51) show that the validity of EMV estimation depends on process accuracy and noise covariance measurement. Filtering accuracy also relies on noise statistics. In addition, even though (42) and (51) theoretically give adaptive EMVs, they bring computational burdens.

5. Application to Attitude Estimation

In this section, we apply the proposed method to a simulated attitude estimation system and compare it with various filters presented in the cited literature. Consider a rotation rigid body without translation that contains a triaxial gyroscope offering angular velocity $\omega_m \in \mathbb{R}^3$ at a 100 Hz sampling frequency, a triaxial accelerometer providing acceleration $f_b \in \mathbb{R}^3$ at 10 Hz sampling frequency, and a triaxial magnetometer measuring magnetic field $m_b \in \mathbb{R}^3$ at 10 Hz sampling frequency. For the spacecraft attitude estimation problem, one can replace accelerometers and magnetometers with sun sensors and magnetometers [23]. The attitude kinematics in terms of the unit quaternion is described by

$$\begin{cases} \dot{q} = \frac{1}{2} q \otimes (\omega_m - b_\omega - n_\omega) \\ \dot{b}_\omega = n_{b_\omega} \end{cases}. \tag{52}$$

Let accelerometer and magnetometer outputs be observations; observation equations are expressed by [23,42]

$$\begin{cases} f_b = q^{-1} \otimes f_n \otimes q + v_f \\ m_b = q^{-1} \otimes m_n \otimes q + v_m \end{cases}, \tag{53}$$

in which $q \in SO(3)$ is the unit quaternion representing the attitude in the body frame with respect to the navigation frame; $b_\omega \in \mathbb{R}^3$ denotes gyroscope bias; $n_\omega \in \mathbb{R}^3$ and $n_{b_\omega} \in \mathbb{R}^3$ are uncorrelated white Gaussian noise; $f_n \in \mathbb{R}^3$ is the gravity vector in the navigation frame; and $m_n \in \mathbb{R}^3$ represents Earth’s magnetic field; and $v_f \in \mathbb{R}^3$ and $v_m \in \mathbb{R}^3$ are sensor noises assumed to be white Gaussian noise. Let $\chi = (q, b_\omega)$, which belongs to matrix Lie group $G = SO(3) \times \mathbb{R}^3$, and with the definition of $\phi(\hat{\chi}, \eta)$ in Section 2, we define the left and right estimation errors as follows:

$$\begin{pmatrix} \tilde{q}_R \\ \tilde{b}_R \end{pmatrix} = \exp_G(-\eta_R) = \begin{pmatrix} \hat{q} \otimes q^{-1} \\ q \otimes (\hat{b}_\omega - b_\omega) \otimes q^{-1} \end{pmatrix} \text{ (right estimation errors)} \tag{54}$$

$$\begin{pmatrix} \tilde{q}_L \\ \tilde{b}_L \end{pmatrix} = \exp_G(\eta_L) = \begin{pmatrix} \hat{q}^{-1} \otimes q \\ b_\omega - \hat{b}_\omega \end{pmatrix} \text{ (left estimation errors)} \tag{55}$$

Then, retractions corresponding with the above estimation errors are

$$\phi_R(\hat{\chi}, -\eta_R) = \exp_G(-\eta_R) \hat{\chi} = \begin{pmatrix} \exp_q(-\eta_{R_{1:3}}) \otimes \hat{q} \\ q^{-1} \otimes (\hat{b}_\omega - \eta_{R_{4:6}}) \otimes q \end{pmatrix} \tag{56}$$

$$\phi_L(\hat{\chi}, \eta_L) = \hat{\chi} \exp_G(\eta_L) = \begin{pmatrix} \hat{q} \otimes \exp_q(\eta_{L1:3}) \\ \hat{b}_\omega + \eta_{L4:6} \end{pmatrix} \quad (57)$$

Numerical Simulations

To evaluate estimation quality, numerical simulations were carried out. In the simulations, we selected the north–east–down frame as the navigation frame. The attitude trajectory that lasted 200 s was generated by $\omega = [-0.1 \cos(0.15t), 0.1 \sin(0.1t), -0.1 \cos(0.05t)]^T$ rad/s. The true trajectory without noise and bias is shown in Figure 2. Gyroscope bias was $\mathbf{b}_\omega = [0.012, -0.021, 0.014]^T$ rad/s, and gyroscope noise was white Gaussian noise subjected to $\mathcal{N}(0, (10^{-3} \text{rad/s})^2)$. Two known corresponding measurement vectors in the navigation frame are $\mathbf{f}_n = [0, 0, g]^T$ and $\mathbf{m}_n = [0.3197, 0, 0.4226]^T G$, where $g = 9.78 \text{ m/s}^2$ is the gravitational acceleration. The noises of the accelerometer and the magnetometer are white Gaussian noise and follow $\mathbf{v}_f \sim \mathcal{N}(0, (2 \times 10^{-3} g)^2)$ and $\mathbf{v}_m \sim \mathcal{N}(0, (4 \times 10^{-3} G)^2)$, respectively. Some sensor parameters are provided by [42]. For filtering, we set the initial covariance matrix as $\mathbf{P}_0 = \text{blkdiag}(10^{-3} \mathbf{I}_{3 \times 3}, 10^{-7} \mathbf{I}_{3 \times 3})$ and the initial gyroscope bias as $\hat{\mathbf{b}}_{\omega 0} = [0, 0, 0]^T$ rad/s. The kernel scale and kernel length that constituted the kernel parameters were selected to be 0.001 and 0.3.

Three cases were considered in our simulations, and in the first two cases, we utilized the trajectory defined above but with different sets of initial attitudes. Case 1 randomly selected the initial attitude from uniform distribution between -10° and 10° , and Case 2 randomly chose the initial attitude from uniform distribution $[-90^\circ, 90^\circ]$. In Case 3, we used the angular velocity set above, but while $50 \text{ s} < t \leq 60 \text{ s}$, angular velocity was set to be $\omega = [-5, 0.5, -0.5]^T$ rad/s; while $100 \text{ s} < t \leq 108 \text{ s}$, angular velocity was set to be $\omega = [-0.1, 0.1, -3]^T$ rad/s; while $150 \text{ s} < t \leq 152 \text{ s}$, angular velocity was set to be $\omega = [0.1, -5, 0.1]^T$ rad/s. Initial attitudes of Case 3 were the same as those in Case 1.

For each case, we conducted 100 independent Monte Carlo simulations and utilized the following root-mean-square error (RMSE) and averaged RMSE of the Euler angles (roll, pitch, yaw) to evaluate estimation quality:

$$\text{RMSE}_{att}(k) = \sqrt{\frac{1}{M} \sum_{m=1}^M \|x_k^m - \hat{x}_k^m\|^2}, \quad (58)$$

$$\text{ARMSE}_{att}(k) = \sqrt{\frac{1}{MK} \sum_{k=1}^K \sum_{m=1}^M \|x_k^m - \hat{x}_k^m\|^2}, \quad (59)$$

where M is the total number of simulations; K is the simulation time step; x_k^m and \hat{x}_k^m represent the truth Euler angles(simulated) and the estimated Euler angles, respectively.

With the simulated system and the initial conditions above, we compared nine different filters on Monte Carlo simulations:

- * $SO(3)$ -based CKF ($SO(3)$ CKF) that considers the attitude embedded in a special orthogonal group ($SO(3)$) and the gyroscope bias in a vector space;
- * right invariant extended Kalman filter (Right-IEKF) and left invariant extended Kalman filter (Left-IEKF) in [21,23], where the gyroscope bias was treated as part of the Lie group structure;
- * right and left cubature Kalman filters on Lie groups (Right-CKF-LG and Left-CKF-LG) [36], which can be treated as extensions of UKF-LG in [16];
- * Bayes–Sard quadrature cubature Kalman filter ($SO(3)$ BSCKF) derived from [11] by utilizing the same Lie group action as the $SO(3)$ CKF;
- * our proposed right and left BSCKFs on Lie groups (Right-BSCKF-LG and Left-BSCKF-LG) in Section 4;

- * the proposed adaptive right BSCKF-LG with time-varying EMVs (Right-BSCKF-LG-adaptive).

We set the sliding window size in both the propagation and update steps of the Right-BSCKF-LG-adaptive to be $n = 35$ in Cases 1 and 2, and $n = 15$ in Case 3.

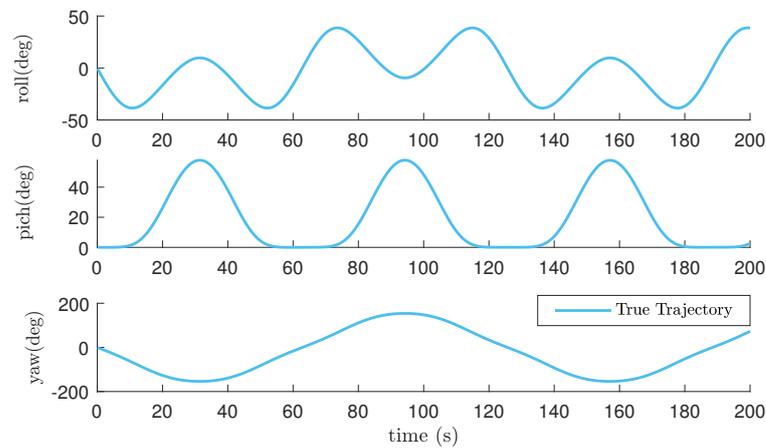


Figure 2. True trajectory of Euler angles in Cases 1 and 2.

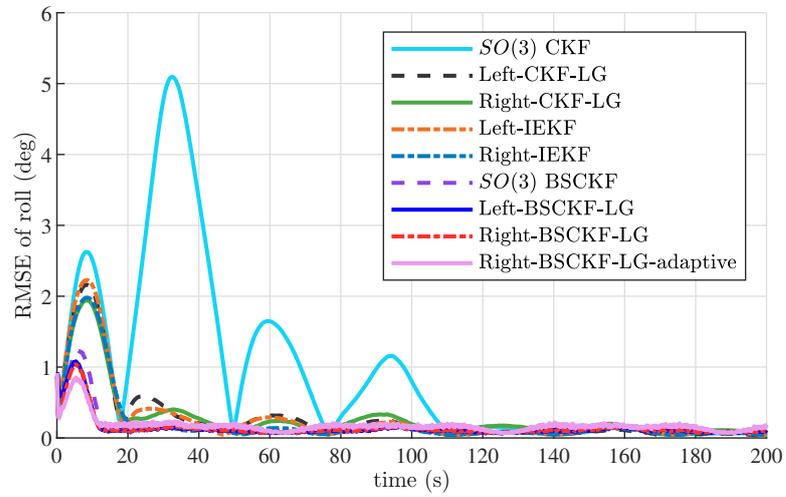
Simulation results of Case 1 are illustrated in Figure 3 and Table 1. Figure 3 compares the RMSEs of the Euler angles, while Table 1 summarizes the corresponding ARMSEs. With small initial attitude errors, the proposed BSCKF-type filters with cubature points performed better than other filters did and needed less time to reach a steady state. Right-BSCKF-LG and its adaptive version obviously outperformed the other filters. Simulation results also verified that the performance of Right-IEKF and Right-CKF-LG was similar and slightly better than that of Left-IEKF and Left-CKF-LG under small initial estimation errors. The selection of sliding window size could affect the estimation accuracy of Right-BSCKF-LG-adaptive, and the filtering accuracy of Right-BSCKF-LG-adaptive was not significantly better than that of Right-BSCKF-LG.

Table 1. ARMSEs of Euler angles in Case 1.

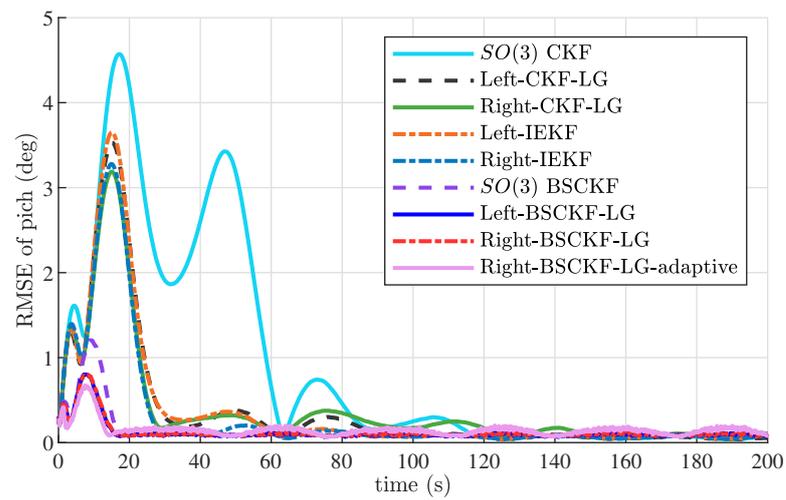
Filters	ARMSEs of Roll (deg)	ARMSEs of Pitch (deg)	ARMSEs of Yaw (deg)
SO(3) CKF	1.5172	1.5174	3.5564
Left-CKF-LG	0.5011	0.8032	2.2217
Right-CKF-LG	0.4571	0.7298	1.9835
Left-IEKF	0.5103	0.8185	2.1791
Right-IEKF	0.4463	0.7226	1.9114
SO(3) BSCKF	0.2505	0.2351	0.7080
Left-BSCKF-LG	0.2156	0.1701	0.5407
Right-BSCKF-LG	0.2062	0.1708	0.5277
Right-BSCKF-LG-adaptive	0.2113	0.1745	0.4586

Case 2 estimation results under large initial attitude errors are demonstrated in Figure 4 and Table 2. These graphs show that the estimation quality of SO(3) CKF, Left-CKF-LG, and Left-IEKF was poor, especially that of SO(3) CKF. In contrast, Right-CKF-LG and Right-IEKF achieved better accuracy and robustness than those of Left-CKF-LG and Left-IEKF. Again, BSCKF-type filters performed better. The estimation accuracy of Right-BSCKF-LG and Right-BSCKF-LG-adaptive was better than that of the other filters. Moreover, the accuracy of Right-BSCKF-LG-adaptive improved compared with that of Right-BSCKF-LG. Right-BSCKF-LG-adaptive performed best in the yaw angle and achieved the fastest convergence rate. As seen from the above analyses, the right estimation errors

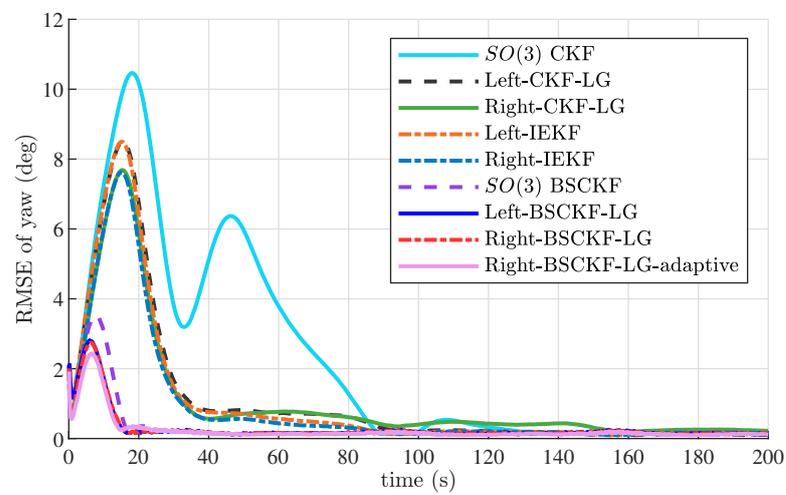
on Lie group $G = SO(3) \times \mathbb{R}^3$ are more suitable for attitude estimation with vector measurements.



(a)

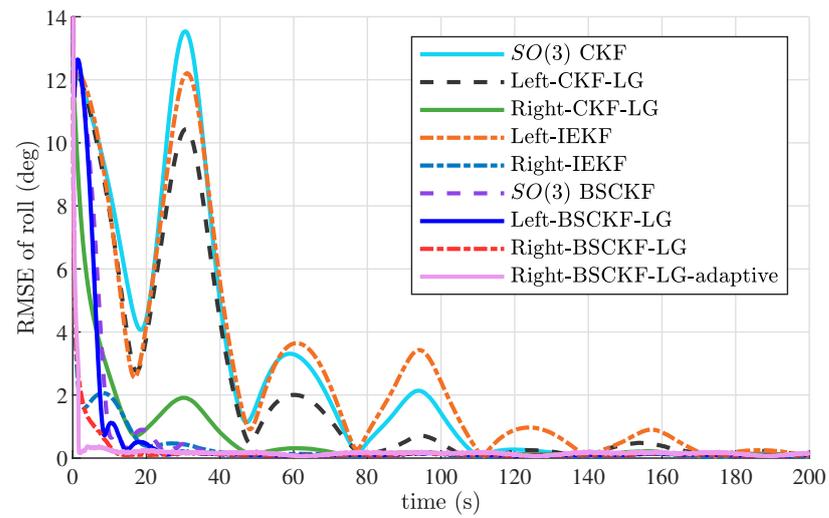


(b)

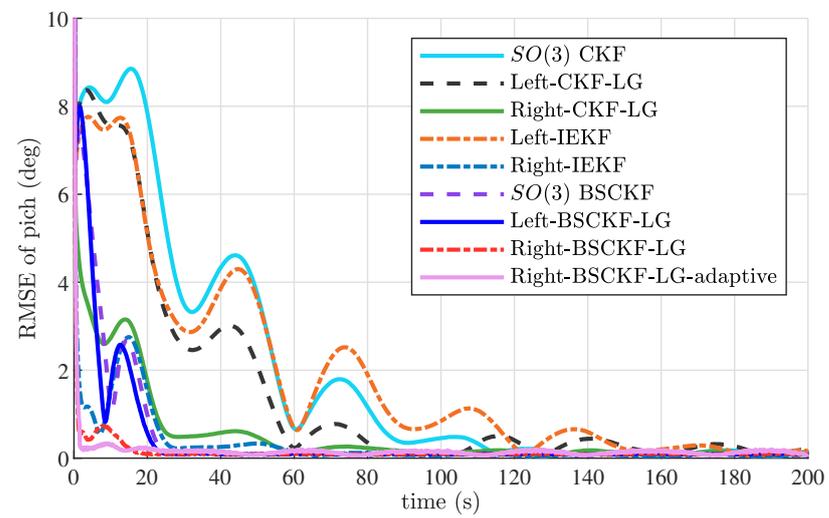


(c)

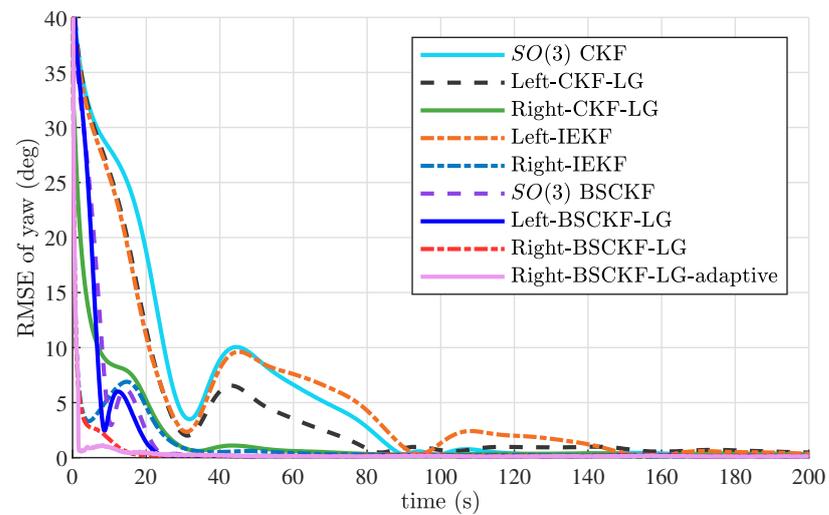
Figure 3. RMSEs of Euler angles in Case 1. (a) RMSEs of roll angle; (b) RMSEs of pitch angle; (c) RMSEs of yaw angle.



(a)



(b)



(c)

Figure 4. RMSEs of Euler angles in Case 2. (a) RMSEs of roll angle; (b) RMSEs of pitch angle; (c) RMSEs of yaw angle.

Table 2. ARMSEs of Euler angles in Case 2.

Filters	ARMSEs of Roll (deg)	ARMSEs of Pitch (deg)	ARMSEs of Yaw (deg)
SO(3) CKF	4.3881	3.2943	10.104
Left-CKF-LG	3.7219	2.6936	8.8371
Right-CKF-LG	1.5143	1.1443	4.1851
Left-IEKF	4.2617	2.9011	9.2021
Right-IEKF	0.7231	0.8164	2.4947
SO(3) BSCKF	1.9847	1.3373	5.6511
Left-BSCKF-LG	1.8803	1.2971	5.4425
Right-BSCKF-LG	0.6037	0.5515	1.8587
Right-BSCKF-LG-adaptive	0.5401	0.4604	1.4699

In Case 3, we assumed that some fast motion occurred in the system trajectory. The true trajectory of Case 3 is shown in Figure 5. Estimation results under small initial attitude errors are indicated in Figure 6 and Table 3. Figure 6 shows that the performance of all filters was worse compared to that in Case 1, partly because of the definition domains of Euler angles. However, the right and left BSCKF-LGs proposed in this paper were still outperformed among all filters, especially adaptive right BSCKF-LG. Table 3 summarizes the ARMSEs of Euler angles. Left BSCKF-LG was slightly more accurate than right BSCKF-LG in this case. The accuracy of Right-BSCKF-LG-adaptive was also greatly improved compared with that of Right-BSCKF-LG.

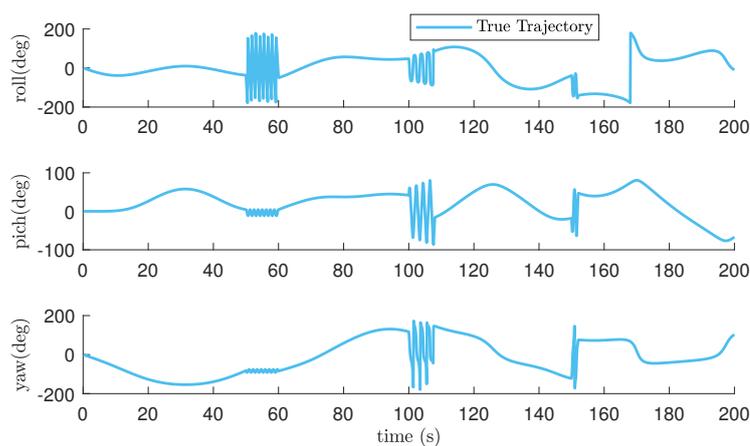
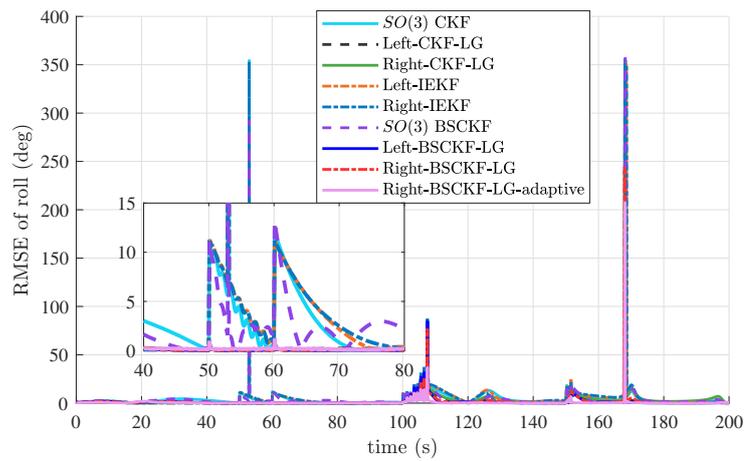


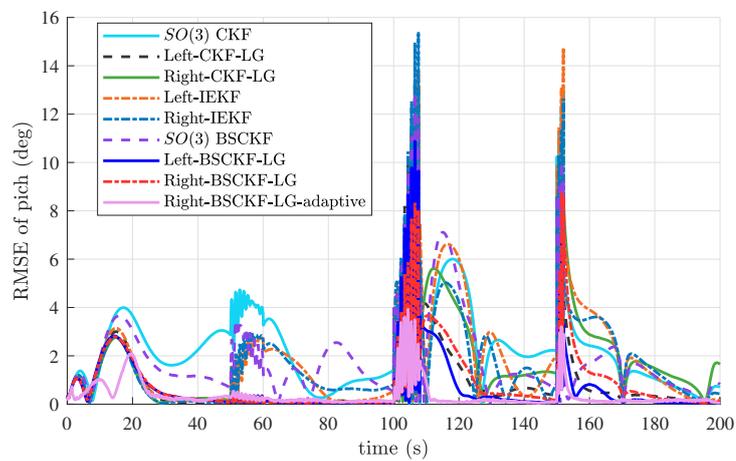
Figure 5. True trajectory of Euler angles in Case 3.

Table 3. ARMSEs of Euler angles in Case 3.

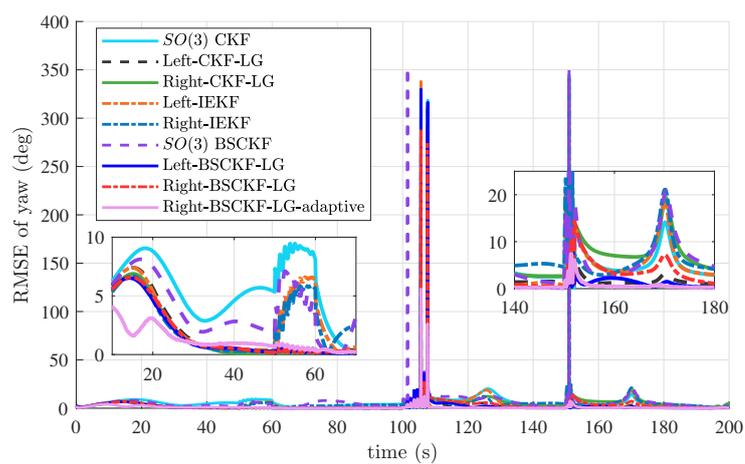
Filters	ARMSEs of Roll (deg)	ARMSEs of Pitch (deg)	ARMSEs of Yaw (deg)
SO(3) CKF	18.5315	2.8001	15.7173
Left-CKF-LG	9.2549	1.4878	12.3105
Right-CKF-LG	15.2969	2.1041	13.0327
Left-IEKF	21.5772	2.6352	13.7434
Right-IEKF	21.6621	2.3658	15.1156
SO(3) BSCKF	14.3881	2.4143	17.4939
Left-BSCKF-LG	3.7793	1.2411	12.2423
Right-BSCKF-LG	8.4749	1.4365	10.6021
Right-BSCKF-LG-adaptive	3.7641	0.5966	2.4787



(a)



(b)



(c)

Figure 6. RMSEs of Euler angles in Case 3. (a) RMSEs of roll angle; (b) RMSEs of pitch angle; (c) RMSEs of yaw angle.

Figure 7 exhibits the computation times of the compared filters. CKF-LGs and BSCKF-LGs spent more computation time, while IEKFs spent minimal time. When the Bayes–Sard weights were calculated in advance according to the known system-state and measurement

dimensions, the computation costs of BSCKF-LGs are almost equal to those of CKF-LGs. Right-BSCKF-LG-adaptive did not significantly increase the computational burden compared with Right-BSCKF-LG. Overall, BSCKF-LGs showed more accurate and robust estimation results in this attitude estimation example.

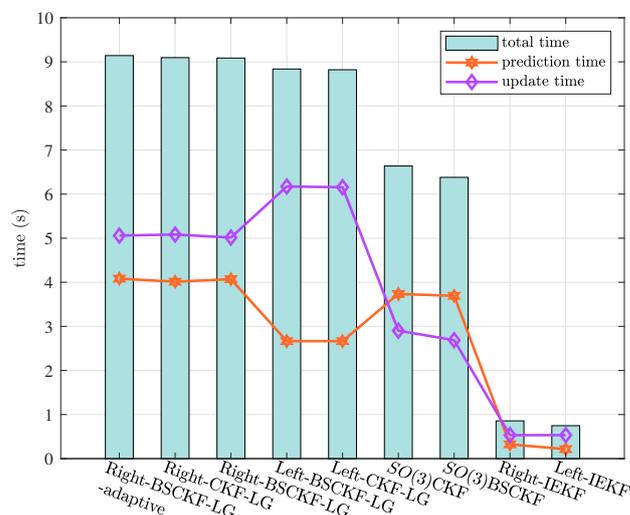


Figure 7. Execution times of different filters.

6. Conclusions

This article proposed a generalized Bayesian estimation algorithm on matrix Lie groups to calibrate numerical integration errors in classical sigma point filters on Lie groups using BSQMT theory. This Bayesian estimation is applicable to both measurements in Euclidean space and those evolving on the Lie group. Afterwards, with the proposed Lie group Bayesian estimator, we presented a Bayes–Sard cubature Kalman filter on Lie groups that comes in two variants. To obtain more accurate estimation, we then developed an approach to calculate adaptive EMVs. Numerical simulation results on quaternion attitude estimation indicated the superiority of our proposed filters over CKFs on Lie groups and unvariant EKFs. Future work includes exploring novel suitable methods to compute adaptive EMVs and applying the proposed filtering algorithm to visual–inertial navigation and fast drone navigation.

Author Contributions: Conceptualization, H.G.; methodology, H.G.; software, H.G.; validation, H.G., H.L. and X.H.; formal analysis, H.G. and Y.Z.; investigation, H.G., X.H. and H.L.; resources, H.L., X.H. and Y.Z.; data curation, H.G. and Y.Z.; writing—original draft preparation, H.G.; writing—review and editing, H.G. and X.H.; visualization, H.G. and Y.Z.; supervision, H.L.; project administration, H.G.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under Grant 62073265.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available because it also forms part of ongoing studies.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof of Lemma 1

Because the dimension of the base function space was $2d$, weights were the unique solver of linear function $\Psi w = \bar{\psi}$. Hence, under the cubature rule:

$$\sum_{n=0}^{2d} w_n q(\xi_n) = \int q(x) \mathcal{N}(x | 0, I) dx$$

where $q(\xi_n) \in \pi$. For standard normal distribution, let $D = 1, \dots, d$; then,

$$\begin{aligned} \int \mathcal{N}(x | 0, I) dx &= \sum_{n=1}^{2d} w_n = 1 \\ \int x_D \mathcal{N}(x | 0, I) dx &= \sum_{n=1}^{2d} w_n x_{n,D} = 0 \\ \int x_D^2 \mathcal{N}(x | 0, I) dx &= \sum_{n=1}^{2d} w_n x_{n,D}^2 = 1 \end{aligned}$$

According to the definition of cubature points, it can be inferred from the above formulas that $w_D = w_{d+D} = 1/2d$. Thus, we obtain Bayes–Sard weights equal to third-degree cubature transform weights.

References

- Aligia, D.A.; Rocca, B.A.; Angelo, C.; Magallán, G.; Gonzalez, G.N. An orientation estimation strategy for low cost IMU using a nonlinear Luenberger observer. *Measurement* **2020**, *173*, 108664. [\[CrossRef\]](#)
- Reina, G.; Messina, A. Vehicle dynamics estimation via augmented extended Kalman filtering. *Measurement* **2018**, *133*, 383–395. [\[CrossRef\]](#)
- Huang, Y.; Zhang, Y.; Bo, X.; Wu, Z.; Chambers, J. A new outlier-robust student's t based Gaussian approximate filter for cooperative localization. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 2380–2386. [\[CrossRef\]](#)
- Kalman, R. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [\[CrossRef\]](#)
- Sunahara Y, Y.K. An approximate method of state estimation for non-linear dynamical systems with state-dependent noise. *Int. J. Control* **1970**, *11*, 957–972. [\[CrossRef\]](#)
- Julier, S.; Uhlmann, J.; Durrant-Whyte, H.F. A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Autom. Control* **2001**, *45*, 477–482. [\[CrossRef\]](#)
- Arasaratnam, I.; Haykin, S.; Elliott, R.J. Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature. *Proc. IEEE* **2007**, *95*, 953–977. [\[CrossRef\]](#)
- Arasaratnam, I.; Haykin, S.; Hurd, T.R. Cubature Kalman filtering for continuous-discrete systems: Theory and simulations. *IEEE Trans. Signal Process.* **2010**, *58*, 4977–4993. [\[CrossRef\]](#)
- Prüher, J.; Šimandl, M. Bayesian quadrature in nonlinear filtering. In Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics, Colmar, France, 21–23 July 2015; pp. 380–387.
- Pruher, J.; Straka, O. Gaussian process quadrature moment transform. *IEEE Trans. Autom. Control* **2017**, *63*, 2844–2854. [\[CrossRef\]](#)
- Pruher, J.; Karvonen, T.; Oates, C.J.; Straka, O.; Sarkka, S. Improved calibration of numerical integration error in Sigma-point filters. *IEEE Trans. Autom. Control* **2020**, *66*, 1286–1296. [\[CrossRef\]](#)
- Bourmaud, G.; Mégret, R.; Giremus, A.; Berthoumieu, Y. From intrinsic optimization to iterated extended Kalman filtering on Lie groups. *J. Math. Imaging Vis.* **2016**, *55*, 284–303. [\[CrossRef\]](#)
- Hauberg, S.; Lauze, F.; Pedersen, K.S. Unscented Kalman filtering on Riemannian manifolds. *J. Math. Imaging Vis.* **2013**, *46*, 103–120. [\[CrossRef\]](#)
- Menegaz, H.; Ishihara, J.Y.; Kussaba, H. Unscented Kalman filters for Riemannian state-space systems. *IEEE Trans. Autom. Control* **2019**, *64*, 1487–1502. [\[CrossRef\]](#)
- Brossard, M.; Barrau, A.; Bonnabel, S. A code for unscented Kalman filtering on Manifolds (UKF-M). In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 1487–1502.
- Brossard, M.; Condomines, J.P. Unscented Kalman filtering on Lie groups. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2485–2491.
- Brossard, M.; Bonnabel, S.; Barrau, A. Unscented Kalman filter on Lie groups for visual inertial odometry. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 649–655.
- Bonnabel, S. Left-invariant extended Kalman filter and attitude estimation. In Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 1027–1032.
- Bonnabel, S.; Martin, P.; Salaun, E. Invariant extended Kalman filter: Theory and application to a velocity-aided attitude estimation problem. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009; pp. 1297–1304.
- Barrau, A.; Bonnabel, S. Intrinsic filtering on Lie groups with applications to attitude estimation. *IEEE Trans. Autom. Control* **2015**, *60*, 436–449. [\[CrossRef\]](#)

21. Barrau, A.; Bonnabel, S. The invariant extended Kalman filter as a stable observer. *IEEE Trans. Autom. Control* **2017**, *62*, 1797–1812. [[CrossRef](#)]
22. Barrau, A.; Bonnabel, S. Invariant Kalman filtering. *Annu. Rev. Control Robot. Auton. Syst.* **2018**, *1*, 237–257. [[CrossRef](#)]
23. Gui, H.; de Ruiter, A.H.J. Quaternion invariant extended Kalman filtering for spacecraft attitude estimation. *J. Guid. Control Dyn.* **2018**, *41*, 863–878. [[CrossRef](#)]
24. Condomines, J.P.; Hattenberger, G. Nonlinear state estimation using an invariant unscented Kalman filter. In Proceedings of the AIAA Guidance, Navigation, and Control, Boston, MA, USA, 19–22 August 2013; p. 4869.
25. Condomines, J.P.; Seren, C.; Hattenberger, G. Invariant unscented Kalman filter with application to attitude estimation. In Proceedings of the IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia, 12–15 December 2017; pp. 2783–2788.
26. Condomines, J.P.; Hattenberger, G. Invariant Unscented Kalman Filtering: A Parametric Formulation Study for Attitude Estimation. 2019. hal-02072456. Available online: <https://hal-enac.archives-ouvertes.fr/hal-02072456> (accessed on 4 April 2022).
27. Barrau, A.; Bonnabel, S. Three examples of the stability properties of the invariant extended Kalman filter. *IFAC-PapersOnLine* **2017**, *50*, 431–437. [[CrossRef](#)]
28. Barrau, A.; Bonnabel, S. An EKF-SLAM algorithm with consistency properties. *arXiv* **2015**, arXiv:1510.06263.
29. Bourmaud, G.; Mégret, R.; Giremus, A.; Berthoumieu, Y. Discrete extended Kalman filter on lie groups. In Proceedings of the 21st European Signal Processing Conference, Marrakech, Morocco, 9–13 September 2013; pp. 1–5.
30. Bourmaud, G.; Mégret, R.; Arnaudon, M.; Giremus, A. Continuous-Discrete extended Kalman filter on matrix Lie groups using concentrated Gaussian distributions. *J. Math. Imaging Vis.* **2015**, *51*, 209–228. [[CrossRef](#)]
31. Phogat, K.S.; Chang, D.E. Invariant extended Kalman filter on matrix Lie groups. *Automatica* **2020**, *114*, 108812. [[CrossRef](#)]
32. Phogat, K.S.; Chang, D.E. Discrete-time invariant extended Kalman filter on matrix Lie groups. *Int. J. Robust Nonlinear Control* **2020**, *30*, 4449–4462. [[CrossRef](#)]
33. Teng, Z.; Wu, K.; Su, D.; Huang, S.; Dissanayake, G. An Invariant-EKF VINS algorithm for improving consistency. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1578–1585.
34. Brossard, M.; Bonnabel, S.; Barrau, A. Invariant Kalman filtering for visual inertial SLAM. In Proceedings of the International Conference on Information Fusion, Cambridge, UK, 10–13 July 2018; pp. 2021–2028.
35. Sjøberg, A.M.; Egeland, O. Lie Algebraic unscented Kalman filter for pose estimation. *IEEE Trans. Autom. Control* **2022**, *in press*. [[CrossRef](#)]
36. Guo, H.; Liu, H.; Zhou, Y.; Li, J. Quaternion invariant cubature Kalman filtering for attitude estimation. In Proceedings of the 2020 3rd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM), Shanghai, China, 4–6 December 2020; pp. 67–72.
37. Barrau, A.; Bonnabel, S. Stochastic observers on Lie groups: A tutorial. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA, 17–19 December 2018; pp. 1264–1269.
38. Karvonen, T.; Oates, C.J.; Srkk, S. A Bayes-Sard cubature method. *arXiv* **2018**, arXiv:1804.03016.
39. Xiong, K.; Liu, L.; Zhang, H. Modified unscented Kalman filtering and its application in autonomous satellite navigation. *Aerosp. Sci. Technol.* **2009**, *13*, 238–246. [[CrossRef](#)]
40. Hu, G.; Gao, B.; Zhong, Y.; Gu, C. Unscented kalman filter with process noise covariance estimation for vehicular ins/gps integration system. *Inf. Fusio* **2020**, *64*, 194–204. [[CrossRef](#)]
41. Wang, J.; Li, M. Covariance regulation based invariant Kalman filtering for attitude estimation on matrix Lie groups. *IET Control Theory Appl.* **2021**, *15*, 2017–2025. [[CrossRef](#)]
42. Nagy, S.B.; Arne, J.T.; Fossen, T.I.; Ingrid, S. Attitude estimation by multiplicative exogenous Kalman filter. *Automatica* **2018**, *95*, 347–355.