Invariant Cubature Kalman Filtering-based visual-inertial odometry for robot pose estimation

Xiaoyue Sang, Jingchao Li, Zhaohui Yuan, Xiaojun Yu*, and Jingqin Zhang, Jianrui Zhang, and Pengfei Yang

***Abstract*—To maintain mechanistic stability while tracking the designated walking route, a robot must be cognizant of employed posture. Generally, visual-inertial odometer (VIO) is utilized for robot state estimation, however, the traditional Cubature Kalman filter VIO (CKF-VIO) cannot transfer rotational uncertainty and compensate for the system's processing error. To effectively improve the accuracy and stability of robot rigid body pose estimation, this paper proposes a matrix Lie group representation-based CKF framework which characterizes the uncertainty prompting in robotic motion while eliminating the VIO system internalization errors. The robot state, consisting of inertial measurement unit (IMU) pose, velocity, and 3D landmarks' positions, is deemed to be a single element of a high-dimensional Lie group $\mathrm{SE}_2 + \mathrm{p}(3)$, while the accelerometers' and gyrometers' biases are appended to the state and estimated as well. The algorithm is validated by simulations with Monte Carlo and experiment. Results show that the CKF-VIO with a high-dimensional Lie group can improve the accuracy and consistency of robot pose estimation.**

***Index Terms*—Vision sensors; IMU; VIO; CKF; state estimation; matrix Lie group**

I. INTRODUCTION

ACCURATE self-position estimation is the basis for the robot control and planning. However, the information obtained through a single positioning sensor is limited, which eventually leads to regressed accuracy and sub-par stability. A visual-inertial odometer (VIO) realizes robot's posture computation by integrating inertial and visual information for its complementary position estimation [1]-[5]. With the development of Micro Electro Mechanical Systems IMU (MEMS IMU), the VIO system has become advantageous for its low power consumption and miniaturization. Currently, VIO is one of the state-of-the-art positioning methods in complex environments without any Global Positioning System (GPS), and now has been widely used in mobile robots, unmanned aerial vehicles, and handheld devices. Due to strong VIO nonlinearity, however, it is difficult to reduce the system state uncertainty, eliminate the accumulation of system process errors, and improve the pose estimation accuracy continuously [6]-[7].

The purpose of VIO is to accurately estimate the state of the mobile robot at each moment by employing filtering or optimizing methods based on the measurement and observed information. Under the Bayesian framework, the optimal solution to the nonlinear filter estimation is to obtain a complete description of the state posterior probability distribution function, however, since the complete description requisites an overflow of parametric calculations and extensive operations, it is infeasible to be implemented in engineering practice. Thus, many suboptimal approximation methods have been proposed, such as the Extended Kalman filter (EKF) [8], particle filter (PF), or unscented Kalman filter (UKF) [9], CKF [10]. In the traditional EKF-VIO algorithm, many feature points have been used as the state quantity, which results in a high filter dimension and a large amount of calculations. The UKF-VIO algorithm employs the basic framework of KF, unequally, it uses first-order linearization and unscented transformation (UT) to approximate the nonlinear functions and posterior of the nonlinear system state, respectively. Although the distribution function improves the state estimation accuracy, its filtering performance is poor or even divergent when dealing with high-dimensional VIO problems. CKF-VIO is also a nonlinear Gaussian filtering-based scheme, and it uses the third-order Spherical-Radial rule to approximate the Gaussian weighted integral. The weight of the cubature sampling point is always positive, and there is no non-positive definite covariance. When dealing with high-dimensional nonlinear filtering problems, the complete numerical stability of CKF is conducive to maintaining a good estimation accuracy, and thus, it is regarded as ideal filtering in the approximate Gaussian integral.

For state estimation, the rotation of a rigid body is one of the main factors that cause the system's nonlinearity. The representation of robot rotation mainly comprises Euler angle, rotation matrix, Quaternion, and Lie algebra, etc. The probability distributions on $SE(3)$, and their role in control and estimation, have been well studied in recent years [11]-[13] [14,15]. Filters that use matrix Lie groups to represent state variables, such as the Invariant EKF (InEKF) [16], have also been designed and used for nonlinear state estimations. The filters combine symmetry preservation theory and EKF and

All authors are from Northwestern Polytechnical University,Xi'an, Shaanxi, 710072, P.R. China,Xiaoyue Sang,sangxykyzq@mail.nwpu.edu.cn
Jingchao Li (e-mail: lijch@mail.nwpu.edu.cn).
Zhaohui Yuan (e-mail: yuanzhh@nwpu.edu.cn).
Xiaojun Yu* (e-mail: xjyu@nwpu.edu.cn).
Jingqin Zhang (e-mail: mrzjq@mail.nwpu.edu.cn).
Jianrui Zhang (e-mail: zhangjr@mail.nwpu.edu.cn).
Pengfei Yang (e-mail: yang_pf@mail.nwpu.edu.cn).

could effectively improve the consistency of system estimation. Specifically, Bonnabel applied InEKF to the SLAM field first [17], while Barrau used the right InEKF in 2D SLAM, and proved its effectiveness in consistency improvement [16]. More recently, InEKF has also been used for the 3D visual-inertial navigation system (VINS) [18]-[19]. However, in the implementation part, landmarks are removed from the state. Heo $et.al.$ proposed right Invariant multistate constraint KF (RI-MSCKF) that combined InEKF with MSCKF, improving the accuracy and consistency of pose estimation in VIO [12]. To reduce the truncation error caused by the linearization of the highly nonlinear function in the InEKF, Brossard $et.al.$ then used the unscented transformation (UT), namely UKF-LG, to achieve accurate pose estimation [13]. From the numerical integration perspective of UT, it could be observed that the stability factor scales linearly with dimension, and thus, significant perturbations would be induced in numerical estimations of moment integrals once the dimension goes beyond four [10]. In addition, it is also impossible to formulate a square-root UKF with numerical advantages like SRCKF. When a negatively weighted sigma point is used to update any matrix, the resulting down-dated matrix may possibly be non-positive definite. Hence, errors may occur when executing a 'pseudo' square-root version of UKF in a limited word-length system [10].

A. Contributions of this paper

To address the problem of robot state estimation in VIO, this paper proposes a right-invariant VIO algorithm, namely, Right-CKF-LG, under the framework of CKF. Right-CKF-LG can be treated as a CKF-based transformation of Right-InEKF [16], yet with the advantage of being more general than Right-InEKF. Right-CKF-LG avoids the complex computation of Jacobians, especially in the Invariant EKF framework, wherein Jacobians are defined with respect to the Lie structure [16]. Right-CKF-LG mainly consists of two components, of which one is the Lie group structure proposed in the field of invariant filtering [16], and the other is the Square Root CKF (SRCKF) [10]. The main contributions of this paper are summarized as follows,

(1) A Right-CKF-LG-based VIO system is proposed and exploited to address the shortcomings of UT, which is unstable in dealing with high-dimensional system states.

(2) The CKF Lie Group (CKF-LG) is applied for state estimation by expanding the state space with the bias vector. The concepts, such as cubature point, state mean, and variance, are extended from Euclidean space to manifold space. The system state variables, including the position, orientation, velocity, and spatial 3D feature points, which are represented by the high-dimensional matrix Lie group, are used to achieve accurate robot pose estimation.

(3) A right invariant VIO algorithm leveraging Lie Group representations is proposed. The formulation of the time and measurement updates of the proposed algorithm are explained in detail.

B. The structure of the paper

The organization of this paper is as follows. Section II describes the mathematical model of VIO, including system dynamics models and measurement models. Section III introduces the Lie group state variables and uncertainty first, and then presents the left-invariant and right-invariant CKF form of the matrix Lie group representation, respectively. Section IV illustrates the detailed Right-CKF-LG for pose estimation in VIO. We have exploited Monte Carlo to simulate the performance of the proposed state estimator in Section V. The EuRoc MAV dataset [20] is used to compare the pose errors of different algorithms in Section VI. Finally, Section VII concludes the whole paper.

II. VIO PROBLEM MODELING

A. Dynamical Model

Most robots are equipped with an IMU, which consists of three orthogonal linear accelerometers and three orthogonal gyroscopes, to measure linear acceleration $\mathbf{a}$ and angular velocity $\boldsymbol{\omega}$. The state to be estimated includes the robot's position $\mathbf{x} \in \mathbb{R}^3$, orientation $\boldsymbol{R} \in SO(3)$, velocity $\mathbf{v} \in \mathbb{R}^3$, IMU bias $\mathbf{b}_{\boldsymbol{\omega}} \in \mathbb{R}^3$, $\mathbf{b}_{\mathbf{a}} \in \mathbb{R}^3$, and the 3D position of landmarks $\mathbf{p}_1, ..., \mathbf{p}_p \in \mathbb{R}^3$ in world frame W. The dynamics of this VIO system could be expressed as,

Body state dynamics,

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{v} \\ \dot{\mathbf{R}} = \mathbf{R}\left(\boldsymbol{\omega} - \mathbf{b}_{\boldsymbol{\omega}} + \mathbf{w}_{\boldsymbol{\omega}}\right)_{\times} \\ \dot{\mathbf{v}} = \mathbf{R}\left(\mathbf{a} - \mathbf{b}_{\mathbf{a}} + \mathbf{w}_{\mathbf{a}}\right) + \mathbf{g} \end{cases} \tag{1}$$

IMU biases dynamics,

$$\begin{cases} \dot{\mathbf{b}}_{\boldsymbol{\omega}} = \mathbf{w}_{\mathbf{b}_{\boldsymbol{\omega}}} \\ \dot{\mathbf{b}}_{\mathbf{a}} = \mathbf{w}_{\mathbf{b}_{\mathbf{a}}} \end{cases} \tag{2}$$

Landmarks dynamics ,

$$\{\dot{\mathbf{p}}_i = 0, i = 1, \ldots, p \tag{3}$$

where the IMU bias dynamics are modeled as typical "Brownian motion", i.e., the derivative is Gaussian white noise to capture the slow time-varying nature of these parameters. $\mathbf{a}$ and $\boldsymbol{\omega}$ are used as noisy and biased inputs to the system. $(\boldsymbol{\omega})_{\times}$ represents the obliquely symmetric matrix related to the cross product of the vectors $\boldsymbol{\omega}$. $\mathbf{g}$ is the acceleration of gravity.

The various white Gaussian continuous time noises can be stacked as,

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_{\boldsymbol{\omega}}^{\mathrm{T}} & \mathbf{w}_{\mathbf{a}}^{\mathrm{T}} & \mathbf{w}_{\mathbf{b}_{\boldsymbol{\omega}}}^{\mathrm{T}} & \mathbf{w}_{\mathbf{b}_{\mathbf{a}}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{4}$$

where $\mathbf{w}$ is centered with autocorrelation $\mathbb{E}(\mathbf{w}(t)\mathbf{w}(s)) = \mathbf{W}\delta(t-s)$.

B. Measurement model

Due to the influence of noises, the process error increases rapidly over time when relying solely on the IMU for pose estimation. To obtain a more accurate robot stature, VIO uses the camera's observation information to update the filter. In addition to IMU measurements, a robot also obtains visual information from the calibrated monocular camera that is rigidly

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2022.3214293

connected with the IMU. The camera observes and tracks the landmarks through a standard pinhole model [25]. The i-th landmark observed through the camera can be demoted as,

$$\mathbf{y}_i = \frac{1}{y_w^i}\left[y_u^i \quad y_v^i\right]^T + \mathbf{n}_{\mathbf{y}}^i, 1 \le i \le p, \tag{5}$$

where $\mathbf{y}_i$ is the normalized pixel position measurement of the landmark, p is the landmark number. $\mathbf{n}_y \sim \mathcal{N}(0, \mathbf{N})$ represents pixel image noise. The landmark is described in the world frame W. To calculate the pixel coordinates of its projection, it is necessary to first convert the landmark to the camera frame C through coordinate transformation,

$$\mathbf{y}_i = \left[\begin{array}{ccc} y_u^i & y_v^i & y_w^i \end{array}\right]^T = \mathbf{R}_{\mathrm{C}}^T\left(\mathbf{R}^T\left(\mathbf{p}_i - \mathbf{x}\right) - \mathbf{x}_{\mathrm{C}}\right) \tag{6}$$

where the term on the right corresponds to the distance from the landmark to the camera represented in C. $\mathbf{R}_C$ is the known rotation matrix, $\mathbf{x}_C$ is the known translation mapping from the body frame to the C.

C. Robot state estimation problem

According to the probabilistic dynamical model shown by Eqs. (1)-(3) and the visual landmark measurement model denoted by Eq. (5), our main objective of robot state estimation is to compute the state probability distributions in a high-dimensional system.

III. CKF ON LIE GROUP

A. Matrix Lie groups and their random variables

We assume that a matrix Lie group is $\mathcal{G} \subset \mathbb{R}^{N\times N}$, which is a subset of square invertible matrices. Lie algebra $\mathfrak{g}$ is related to Lie groups. And we define the exponential mapping $\mathbb{R}^{dim\mathfrak{g}} \to \mathcal{G}$ of the Lie group in Eq.(7), $\log : \mathcal{G} \to \mathbb{R}^q$, which leads to Eq.(8),

$$\exp(\boldsymbol{\xi}) = \exp_m\left(\boldsymbol{\xi}^\wedge\right) \tag{7}$$

$$\log(\exp(\boldsymbol{\xi})) = \boldsymbol{\xi} \tag{8}$$

where $\exp_m(\cdot)$ is the exponential mapping of the n×n matrix. Matrix Lie groups can be represented in Euclidean space $\mathbb{R}^q$ by exponential mapping, $q = dim(\mathcal{G})$. In fact, for any element $\boldsymbol{\chi} \in \mathcal{G}$, the tangent space vector at that point corresponds to it, called a Lie algebra, to associate a matrix $\mathcal{L}_{\mathfrak{g}}(\xi) = \xi^\wedge$ in the tangent space at $\mathbf{I}$. Consequently, for random variables in Lie groups, we use the corresponding Lie algebra exponential mapping to describe the uncertainty [14]. $\boldsymbol{\xi}$ is the bias factor describing the uncertainty of a variable.

B. CKF-LG Description

By representing the state error as a variable of a Lie algebra $\boldsymbol{\xi}$, we can establish a left-invariant CKF and a right-invariant CKF for any state that evolves on a Lie group, both filters obey the uncertainty representation described above.

1)Right-CKF-LG: The state is modeled as $\mathbf{X}_n \sim \mathcal{N}_R\left(\bar{\mathbf{X}}_n, \mathbf{P}_n\right)$, the state mean is $\bar{\mathbf{X}}_n$, and the deviation is $\boldsymbol{\xi} \sim \mathcal{N}\left(0, \mathbf{P}_n\right)$, i.e. using the Right-invariant form of uncertainty [13], for more details, please see [13]. In particular, the consistency properties of EKF-VIO take a right-invariant form, see [18]. Therefore, we have only demonstrated the Right-invariant form in detail.

2)Left-CKF-LG: Similarly, the state is modeled as $\mathbf{X}_n \sim \mathcal{N}_L\left(\bar{\mathbf{X}}_n, \mathbf{P}_n\right)$, i.e., the left-invariant form of uncertainty [13].

During the state transfer process, when the estimated state contains not only a matrix $\boldsymbol{\chi}$ belonging to the Lie group but also the IMU bias vector $\mathbf{b}$, we can easily apply CKF-LG by expanding the state space, i.e., the bias vector is attached to the state, $(\boldsymbol{\chi}, \mathbf{b})$. $\mathbf{b}$ can be represented as $\mathbf{b} = \bar{\mathbf{b}} + \tilde{\mathbf{b}}$, $\bar{\mathbf{b}}$ and $\tilde{\mathbf{b}}$ being the mean and a random deviation, respectively. Therefore, the mean of the state variables of the whole system is $\bar{\mathbf{X}} = (\bar{\boldsymbol{\chi}}, \bar{\mathbf{b}})$, and the random deviation vector is $\boldsymbol{\sigma} = (\boldsymbol{\xi}, \tilde{\mathbf{b}})$. $\bar{\boldsymbol{\chi}} \in \mathcal{G}$ represents the mean value of $\boldsymbol{\chi}$. The Lie group variables are processed by the CKF-LG method, while the IMU bias vector is processed by the traditional CKF method. The filter consists of two steps: time update and measurement update, as shown in Algorithm 1.

Algorithm 1: Left and Right CKF on Lie groups

input : $\bar{\boldsymbol{\chi}}$, $\bar{\mathbf{b}}$, $\bar{\mathbf{X}}$,**u**,**Q**,**Y**,**N**;
1 $\bar{\boldsymbol{\chi}}, \bar{\mathbf{b}}, \bar{\mathbf{X}} \leftarrow Propagation(\bar{\boldsymbol{\chi}}, \bar{\mathbf{b}}, \bar{\mathbf{X}}, \mathbf{u}, \mathbf{Q})$;
2 if received measurement then;
3 $\bar{\boldsymbol{\chi}}, \bar{\mathbf{b}} \leftarrow Update(\bar{\boldsymbol{\chi}}, \bar{\mathbf{b}}, \mathbf{Y}, \mathbf{N})$;
output: $\bar{\boldsymbol{\chi}}$, $\bar{\mathbf{b}}$;

IV. RIGHT-CKF-LG FOR POSE ESTIMATION

The Right-CKF-LG constructs the estimated state $(\mathbf{x}, \mathbf{R}, \mathbf{v}, \mathbf{b}_{\boldsymbol{\omega}}, \mathbf{b}_{\mathbf{a}}, \mathbf{p}_1, ..., \mathbf{p}_p)$ into a high-dimensional special orthogonal group matrix, selects the cubature points using the third-order Spherical-Radial rule, and transfers the state mean and variance through cubature transformation, which are computed in the tangential space of the manifold. This section introduces the time update and measurement update steps to transfer the uncertainty of Lie group variables for realizing the pose estimation of robot.

A. Time Discrete

To apply CKF to a Lie group, it is imperative to discretize the dynamics, and the state space must be (partially) embedded in a matrix Lie group. We consider a discrete-time dynamical system as

$$\mathbf{X}_{n+1} = f\left(\mathbf{X}_n, \mathbf{u}_n, \mathbf{w}_n\right), \tag{9}$$

where $\mathbf{u}_n$ is a known input variable.

Considering discrete measurements of this dynamical system,

$$\mathbf{y}_{n+1} = h\left(\mathbf{X}_n, \mathbf{n}_n\right). \tag{10}$$

$\mathbf{w}_n \sim \mathcal{N}_R\left(\bar{\mathbf{X}}_n, \mathbf{Q}_n\right)$ and $\mathbf{n}_n \sim \mathcal{N}\left(0, \mathbf{N}_n\right)$ are Gaussian white noises.

B. System State Variables

In the robot state estimation process, it is necessary to clarify the state variable structure transmitted by the VIO system. $(\mathbf{x}, \mathbf{R}, \mathbf{v}, \mathbf{p}_1, ..., \mathbf{p}_p)$ are constructed as a high-dimensional

Euclidean group matrix, and the matrix structure of $\chi \in SE_{2+p}(3)$ as

$$\chi = \begin{bmatrix} \mathbf{R} & \mathbf{v} & \mathbf{x} & \mathbf{p}_1 & \cdots & \mathbf{p}_p \\ & \mathbf{0}_{(p+2)\times(3)} & & \mathbf{I}_{(p+2)\times(p+2)} & & \end{bmatrix} \quad (11)$$

The dimension of the Lie group and Lie algebra is $3 + 3(2+p)$. Uncertainty, that is, the corresponding deviation variable is $\left[\boldsymbol{\xi}_{\mathbf{R}}^T \boldsymbol{\xi}_{\mathbf{v}}^T \boldsymbol{\xi}_{\mathbf{x}}^T \boldsymbol{\xi}_{\mathbf{p}_1}^T \cdots \boldsymbol{\xi}_{\mathbf{p}_\mathbf{p}}^T\right]^T \in \mathbb{R}^{3p+9}$, where $\boldsymbol{\xi}_{\mathbf{R}}^T$, $\boldsymbol{\xi}_{\mathbf{v}}^T$ and $\boldsymbol{\xi}_{\mathbf{x}}^T$ are the deviation components corresponding to orientation, velocity and position, respectively. $\boldsymbol{\xi}_{\mathbf{p}_1}^T \cdots \boldsymbol{\xi}_{\mathbf{p}_\mathbf{p}}^T$ are the deviation components corresponding to the positions of p feature points.

Mapping it to Lie algebra by $\mathcal{L}_{\mathfrak{g}} : \xi \mapsto \mathcal{L}_{\mathfrak{g}}(\xi) = \xi^{\wedge}$, the antisymmetric matrix structure corresponding to the vector is,

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}_{\mathbf{R}}^T & \boldsymbol{\xi}_{\mathbf{v}}^T & \boldsymbol{\xi}_{\mathbf{x}}^T & \boldsymbol{\xi}_{\mathbf{p}_1}^T \cdots \boldsymbol{\xi}_{\mathbf{p}_\mathbf{p}}^T \end{bmatrix} \in \mathbb{R}^{3p+9} \quad (12)$$

The exponential mapping expression is,

$$\exp \boldsymbol{\xi} = \mathbf{I} + \boldsymbol{\xi}^{\wedge} + \frac{1-\cos(\|\boldsymbol{\xi}_{\mathbf{R}}\|)}{\|\boldsymbol{\xi}_{\mathbf{R}}\|}\boldsymbol{\xi}^{\wedge 2} + \frac{\|(\xi)_{\mathbf{R}}\| - \sin(\|\boldsymbol{\xi}_{\mathbf{R}}\|)}{\|\boldsymbol{\xi}_{\mathbf{R}}\|^3}\boldsymbol{\xi}^{\wedge 3}. \quad (13)$$

C. Evolving the state in a Lie group

The state can be partially embedded in a Lie group, let $\boldsymbol{\chi}_n$ be the matrix of the Lie group. Using this embedding, the state can be sequentially represented as $(\boldsymbol{\chi}_n, \mathbf{b}_n)$, let the bias vector be $\mathbf{b} = \left[\mathbf{b}_\omega^T, \mathbf{b}_\mathrm{g}^T\right]^T \in \mathbb{R}^6$. The deviation on $\boldsymbol{\chi}_n$ can be calculated in [13] in detail.

Considering input vector as the input vector $\mathbf{u} = \begin{bmatrix} \boldsymbol{\omega}^T & \mathbf{a}^T \end{bmatrix}^T$, uncertainty and discrete-time dynamical model associated with Right-CKF-LG can be portrayed as: Uncertainty,

$$\begin{cases} \boldsymbol{\chi}_n = \exp(\boldsymbol{\xi})\bar{\boldsymbol{\chi}}_n \\ \mathbf{b}_n = \bar{\mathbf{b}}_n + \tilde{\mathbf{b}} \end{cases}, \begin{bmatrix} \boldsymbol{\xi} \\ \tilde{\mathbf{b}} \end{bmatrix} \sim \mathcal{N}(0, \mathbf{P}_n), \quad (14)$$

Dynamics,

$$\{\boldsymbol{\chi}_n, \mathbf{b}_n = f(\boldsymbol{\chi}_{n-1}, \mathbf{u}_n - \mathbf{b}_{n-1}, \mathbf{w}_n), \quad (15)$$

Observations,

$$\begin{cases} \mathbf{Y}_n = \begin{bmatrix} \mathbf{y}_1^T & \cdots & \mathbf{y}_p^T \end{bmatrix}^T := \mathbf{Y}(\boldsymbol{\chi}_n, \mathbf{n}_n) \\ \mathbf{y}_i = \frac{1}{y_w^i}\begin{bmatrix} y_u^i & y_v^i \end{bmatrix}^T + \mathbf{n}_{\mathbf{y}}^i, i = 1, \ldots, p \end{cases}. \quad (16)$$

where $(\bar{\boldsymbol{\chi}}_n, \bar{\mathbf{b}}_n) \in \mathbb{R}^{3p+15}$ represents the mean of the estimated state, $(\mathbf{P}_n) \in \mathbb{R}^{(3p+15)\times(3p+15)}$ is the covariance matrix that defines the state uncertainty $(\boldsymbol{\xi}, \tilde{\mathbf{b}})$. $\mathbf{Y}_n$ contains the observations of landmarks and the associated discrete Gaussian noise $\mathbf{w}_n \sim \mathcal{N}(0, \mathbf{Q})$.

D. Time update

The CKF is used to realize the time update. First, construct the sampling cubature points to describe the distribution of the state variables, transfer them through the IMU motion model, and finally the cubature points are weighted and summed to obtain the transferred state mean and variance. At time k, the mean value of the state variable of the system is $(\bar{\chi}_k, \bar{\mathbf{b}}_k)$, the random deviation vector is $\boldsymbol{\sigma}_k = \begin{bmatrix} \boldsymbol{\xi}_k & \tilde{\mathbf{b}}_k \end{bmatrix}^T$ with the Gaussian distribution of variance $\mathbf{P}_k$, and the square root of the covariance is $\mathbf{S}_k$, satisfies $\mathbf{P}_k = \mathbf{S}_k\mathbf{S}_k^T$.

1) Construct sampling cubature point: The control input of the system is $\mathbf{u}_k = \begin{bmatrix} \boldsymbol{\omega}_k & \mathbf{a}_k \end{bmatrix}$, the corresponding noise variance and the square root of its covariance is $\mathbf{Q}_u$ and $\mathbf{S}_u$. To calculate the sampling cubature point, the system input and its covariance square root factor are augmented,

$$\begin{aligned} \mathbf{X}_{\text{aug}}^k &= [\mathbf{X}_k, \mathbf{u}_k]^T \\ \mathbf{S}_{\text{aug}}^k &= \begin{bmatrix} \mathbf{S}_k & \\ & \mathbf{S}_u \end{bmatrix}, \end{aligned} \quad (17)$$

The dimension of the square root of the covariance $\mathbf{S}_{\text{aug}}$ after augmentation is $l_s = 3p + 21$. According to the cubature point sampling rule, the sample process as

$$\{\zeta^j\} = \sqrt{l_s}\mathbf{S}_{\text{aug}}[1]_j, j = 1, \ldots, 2l_s, \quad (18)$$

The structure of $\boldsymbol{\zeta}$ includes bias vector $\boldsymbol{\xi}_k^j$, IMU bias vector $\tilde{\mathbf{b}}_k^j$ and input noise $\mathbf{n}_u^j$, i.e., $\boldsymbol{\zeta}_i = \left[\boldsymbol{\xi}_k^j, \tilde{\mathbf{b}}_k^j, \mathbf{n}_u^j\right]^T$. According to the cubature law, since the Lie group matrix in the manifold space is not closed to addition, so the $\bigoplus$ is defined, the cubature point is in the form of $\bar{\mathbf{X}}_{aug}\prime \bigoplus \prime\zeta^j$. The IMU zero bias and the input satisfy the Euclidean space addition, $\boldsymbol{\chi}$ adopts the exponential mapping to describe the process of superimposing a deviation component near the mean value. The specific calculation form of the system variable cubature point is,

$$\bar{\mathbf{X}}_{\text{aug},k} \oplus \boldsymbol{\zeta}^j = \begin{cases} \exp\left(\boldsymbol{\xi}_k^j\right)\bar{\boldsymbol{\chi}}_k \\ \bar{\mathbf{b}}_k + \tilde{\mathbf{b}}_k^j \\ \mathbf{u}_k + \mathbf{n}_u^j \end{cases}, j = 1, \ldots, 2l_s, \quad (19)$$

2) Transfer cubature point: The cubature point set is transferred through the IMU motion $f(\cdot)$, and the transferred cubature point set is,

IMU motion equations,

$$f(\cdot) = \begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_k\boldsymbol{\Delta}k + \frac{1}{2}\mathbf{a}_k\boldsymbol{\Delta}k^2 \\ \mathbf{R}_{k+1} = \mathbf{R}_k \cdot \exp(\boldsymbol{\omega}_k\boldsymbol{\Delta}k) \\ \mathbf{v}_{k+1} = \mathbf{v}_k + \mathbf{a}_k\Delta k, \end{cases} \quad (20)$$

$$\begin{bmatrix} \boldsymbol{\chi}_{k+1|k}^j & \mathbf{b}_{k+1|k}^j \end{bmatrix} = f\left(\exp\left(\boldsymbol{\xi}_k^j\right)\overline{\boldsymbol{\chi}}_k, \mathbf{u}_k - \mathbf{b}_k^j, \mathbf{n}_u^j\right), \quad (21)$$

where the structural form of the state cubature point is $\boldsymbol{\chi}_k^j = \exp\left(\boldsymbol{\xi}_k^j\right)\overline{\boldsymbol{\chi}}_k = \begin{bmatrix} \mathbf{R}_k^j & \mathbf{v}_k^j & \mathbf{x}_{I.k} & \mathbf{p}_k^{1,j} & \cdots & \mathbf{p}_k^{p,j} \end{bmatrix}$, and the corresponding noise is $\mathbf{n}_u^j = \begin{bmatrix} \mathbf{n}_\omega^j & \mathbf{n}_\mathbf{a}^j \end{bmatrix}$.

The j cubature point is transferred through the motion equations, and the main update transpires in IMU component.

$$\begin{cases} {}_{\mathrm{I}}^{\mathrm{W}}\mathbf{R}_{k+1|k}^j = {}_{\mathrm{I}}^{\mathrm{W}}\mathbf{R}_k^j \cdot \exp\left[\left({}^{\mathrm{I}}\boldsymbol{\omega}_k - \bar{\mathbf{b}}_{\boldsymbol{\omega},k} - \tilde{\mathbf{b}}_{\boldsymbol{\omega}k}^j + \mathbf{n}_{\boldsymbol{\omega}}^j\right)\boldsymbol{\Delta}k\right], \\ {}^{\mathrm{W}}\mathbf{v}_{k+1/k}^j = {}^{\mathrm{W}}\mathbf{v}_k^j + \left[{}_{\mathrm{I}}^{\mathrm{W}}\mathbf{R}_k^j\left({}^{\mathrm{I}}\mathbf{a}_k - \bar{\mathbf{b}}_{a,k} - \tilde{\mathbf{b}}_{a,k}^j - \mathbf{n}^j\right) + \mathbf{g}\right] \cdot \Delta k \\ {}^{\mathrm{W}}\mathbf{x}_{k+1k}^j = {}^{\mathrm{W}}\mathbf{x}_k^j + {}^{\mathrm{W}}\mathbf{v}_{k+1|k}^j \cdot \Delta k \end{cases} \quad (22)$$

${}_{\mathrm{I}}^{\mathrm{W}}\mathbf{R}$represents the transfer from the I frame to the W frame to describe the pose of the IMU in W frame.

Transferring the mean value of the state variable, the IMU pose and the mean value of deviation at k+1 time are,

$$\begin{bmatrix} \bar{\mathbf{X}}_{k+1|k} & \bar{\mathbf{b}}_{k+1|k} \end{bmatrix} = f\left(\bar{\boldsymbol{\chi}}_k, \mathbf{u}_k - \bar{\mathbf{b}}_k, 0\right), \quad (23)$$

3) Calculation of the square root factor of covariance: In order to calculate the square root of the covariance of the state variable after updating, it is necessary to calculate the transmitted deviation vector, which is $\left[\bar{\xi}_{k+1|k} \quad \tilde{\mathbf{b}}_{k+1|k}\right]^T$. The deviations are,

$$\boldsymbol{\xi}^j_{k+1|k} = \frac{1}{\sqrt{2l_s}} \log\left(\boldsymbol{\chi}^j_{k+1|k}\bar{\boldsymbol{\chi}}^{-1}_{k+1|k}\right), j = 1, \cdots, 2l_s, \quad (24)$$

$$\tilde{\mathbf{b}}^j_{k+1|k} = \frac{1}{\sqrt{2l_s}} \log\left(\mathbf{b}^j_{k+1|k} - \bar{\mathbf{b}}^j_{k+1|k}\right), j = 1, \cdots, 2l_s, \quad (25)$$

By performing $\mathcal{QR}$ decomposition on the matrix composed of the deviation and the noise covariance, the square root of the covariance after updating can be calculated as:

$$\mathcal{QR} \leftarrow \left[\begin{array}{cccccccc} \boldsymbol{\xi}^1_{k+1k} & \cdots & \xi^{2l_s}_{k+1|k}; & \tilde{\mathbf{b}}^1_{k+1k} & \cdots & \tilde{\mathbf{b}}^{2l_s}_{k+1|k}; & \mathbf{0} & \mathbf{S}_u \end{array}\right] \quad (26)$$

$$\mathbf{S}_{k+1|k} = r^T, \quad (27)$$

Algorithm 2: Propagation function for Right-CKF-LG

input : $\bar{\chi}$, $\bar{\mathbf{b}}$,$\mathbf{u}$,$\mathbf{Q}$,$\mathbf{N}$;

$\mathbf{u} \leftarrow (\mathbf{u} - \bar{\mathbf{b}})$;`// unbiased input`

$\boldsymbol{\chi} = \bar{\chi}$;`// save non propagated state`

$\bar{\chi}, \bar{\mathbf{b}} = f(\boldsymbol{\chi}, \mathbf{u}, 0)$;`// propagate mean`

$\bar{\mathbf{X}}_{aug,k'} \oplus \prime\zeta^j \approx \begin{cases} \exp\left(\boldsymbol{\xi}^j_k\right)\bar{\boldsymbol{\chi}}_k \\ \bar{\mathbf{b}}_k + \tilde{\mathbf{b}}^j_k \\ \mathbf{u}_k + \mathbf{n}^j_u \end{cases}, j = 1, \ldots, 2l_s$;`// cubature points generation`

$\boldsymbol{\chi}^j_{k+1|k}, \mathbf{b}^j_{k+1|k} \leftarrow f\left(\exp\left(\boldsymbol{\xi}^j_k\right)\bar{\boldsymbol{\chi}}_k, \mathbf{u}_k - \mathbf{b}^j_k, \mathbf{n}^j_u\right)$;

$\boldsymbol{\xi}^j_{k+1|k} \leftarrow \frac{1}{\sqrt{2l}} \log\left(\boldsymbol{\chi}^j_{k+1|k}\bar{\boldsymbol{\chi}}^{-1}_{k+1|k}\right), j = 1, \cdots, 2l_s$;`// cubature points propagation`

$\mathcal{QR} \leftarrow \left[\begin{array}{cccccccc} \boldsymbol{\xi}^1_{k+1k} & \cdots & \boldsymbol{\xi}^{2l}_{k+1|k} & \tilde{\mathbf{b}}^1_{k+1k} & \cdots & \tilde{\mathbf{b}}^{2l_e}_{k+1|k}; & \mathbf{0} & \mathbf{S}_u \end{array}\right]$.

output: $\bar{\chi}_{k+1|k}$, $\boldsymbol{S}_{k+1|k}$.

E. Measurement update

The observation from feature points is used for the measurement update of the filter. The image feature tracking uses the Kanade-Lucas-Tomasi (KLT) algorithm. Assuming that the number of tracking feature points at two consecutive times is m, the observed structure form is in Eq. (16). Using the cubature transformation method to obtain the observation error of the tracking feature at k+1 time. $\mathbf{X}_{k+1|k}$ and $\mathbf{S}_{k+1|k}$ obtained in the time update are first augmented by $\mathbf{y}_{k+1}$ and the observed covariance square root factor $\mathbf{S}_R$,

$$\begin{aligned} \mathbf{X}^{k+1}_{\text{aug}} &= [\mathbf{X}_{k+1}, 0]^T \\ \mathbf{S}^k_{\text{aug}} &= \left[\begin{array}{cc} \mathbf{S}_{k+1|k} & \\ & \mathbf{S}_V \end{array}\right], \end{aligned} \quad (28)$$

where $\mathbf{S}_V = chol(\mathbf{Q}_V) \in \mathbb{R}^{2m \times 2m}$, $\mathbf{Q}_V$ is the observed variance. The dimension of $\mathbf{S}^{k+1}_{\text{aug}}$ after augmentation is $l_m = 5p + 15$. In order to calculate the cubature point, sampling is performed for $\mathbf{C}^j$,

$$\mathbf{C}^j = \sqrt{l_m} S^{k+1}_{\text{aug}}[1], j = 1, \cdots, 2l_m, \quad (29)$$

where the structure of $\mathbf{C}^j$ is $\left[\begin{array}{ccc} \boldsymbol{\xi}^j_{k+1|k} & \tilde{\mathbf{b}}^j_{k+1k_k} & \mathbf{n}^j_V \end{array}\right]^T, j = 1, \ldots, 2l_s$. Similar to the calculation in the time update, the cubature point calculation process can be presented as:

$$\bar{\mathbf{X}}^{k+1}_{\text{aug},k} \oplus \mathbf{C}^j = \begin{cases} \exp\left(\boldsymbol{\xi}^j_{k+1|1}\right)\bar{\boldsymbol{\chi}}_{k+1|k} \\ \bar{\mathbf{b}}_{k+1|k} + \tilde{\mathbf{b}}^j_{k+1|k} \\ \mathbf{n}^j_V \end{cases}, j = 1, \ldots, 2l_s \quad (30)$$

where $\mathbf{n}_V$ is the observed noise. Transfer cubature point by the observation function Eq.(16), and the observation one is,

$$\mathbf{y}^j_{k+1|k} = h(\exp(\boldsymbol{\xi}^j_{k+1|k})\bar{\boldsymbol{\chi}}^j_{k+1|k}, \mathbf{n}^j_V), \quad (31)$$

At the time k+l, the estimated observation feature $\bar{\mathbf{y}}_{k+1|k}$ and the deviation of each observation cubature point $\mathbf{e}^j$ can be calculated,

$$\bar{\mathbf{y}}_{k+1|k} = \frac{1}{2l_m}\sum_{j=1}^{2l_m} \mathbf{y}^j_{k+1|k}, \quad (32)$$

$$\mathbf{e}^j = \frac{1}{\sqrt{2l_m}}\left(\mathbf{y}^j_{k+1|k} - \bar{\mathbf{y}}_{k+1|k}\right), j = 1, \cdots, 2l_m, \quad (33)$$

By performing $\mathcal{QR}$ decomposition on the matrix formed by $\mathbf{e}^j$ and the square root factor of the observed noise variance $\mathbf{S}_V$, the square root of the observed covariance $\mathbf{S_y}$ is portrayed by Eq. (35).

$$\mathcal{QR} \leftarrow \left[\begin{array}{ccccc} \mathbf{e}^1 & \cdots & \mathbf{e}^{2l_m}; & \mathbf{0} & \mathbf{S}_V \end{array}\right] \quad (34)$$

$$\mathbf{S_y} = r^T, \quad (35)$$

The cross-correlation covariance matrix between the state and the observation is $\mathbf{P}_{X,y}$, and the Kalman Gain in measurement update is ,

$$\mathbf{P}_{X,y} = \frac{1}{2l_m}\sum_{j=1}^{2l_m}\left[\boldsymbol{\xi}^j, \tilde{\mathbf{b}}^j\right]^T\left(\mathbf{y}^j - \bar{\mathbf{y}}\right), \quad (36)$$

$$\mathbf{K} = \mathbf{P}_{X,y}(\mathbf{S}^T_y)^{-1}\mathbf{S}^{-1}_y, \quad (37)$$

The state deviation increment, the updated mean of the state and the square root of the covariance are calculated in Eqs.(38-42).

$$\left[\begin{array}{c} \delta\bar{\boldsymbol{\xi}} \\ \delta\bar{\mathbf{b}} \end{array}\right] = \mathbf{K}\left(\mathbf{y}_{k+1} - \bar{\mathbf{y}}_{k+1|k}\right), \quad (38)$$

$$\boldsymbol{\chi} = \left[\begin{array}{c} \boldsymbol{\chi}_{k+1|k+1} \\ \mathbf{b}_{k+1|k+1} \end{array}\right] = \begin{cases} \exp(\delta\boldsymbol{\xi})\bar{\boldsymbol{\chi}}_{k+1|k+1} \\ \bar{\mathbf{b}}_{k+1|k+1} + \delta\mathbf{b} \end{cases} \quad (39)$$

$$\mathcal{QR} \leftarrow [\left[\begin{array}{c} \boldsymbol{\xi}^j_{k+1|k} \\ \tilde{\mathbf{b}}^j_{k+1|k} \end{array}\right] - \mathbf{K}\mathbf{e}^j; \quad \mathbf{0} \quad \mathbf{S}_V] \quad (40)$$

$$\mathbf{S}_{k+1|k+1} = r^T, \quad (41)$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{S}_{k+1|k+1}\mathbf{S}^T_{k+1|k+1} \quad (42)$$

Thus, the VIO system realizes the pose estimation of the robot through the CKF. The cubature transformation method

Algorithm 3: Update function for the Right-CKF-LG

input : $\bar{\chi}$, $\bar{\mathbf{b}}$,$\mathbf{Y}$,$\mathbf{Q}$,$\mathbf{N}$;

$\mathbf{Y_0} = \mathbf{Y}(\bar{\chi}, \mathbf{0})$;`// see (21) and (5-6)`

$\bar{\mathbf{X}}^{k+1}_{aug,k}{}' \oplus{}' \mathbf{C}^j \approx \begin{cases} \exp\left(\boldsymbol{\xi}^j_{k+1|k}\right)\bar{\chi}_{k+1|k} \\ \bar{\mathbf{b}}_{k+1|k} + \tilde{\mathbf{b}}^j_{k+1|k} \\ \mathbf{y}_{k+1} + \mathbf{n}^j_V \end{cases}, j = 1, \ldots, 2l_s$;`// cubature points generation`

$\chi_j = \exp(\boldsymbol{\xi})\,\bar{\chi}$;

$\boldsymbol{y}^j_{k+1|k} \leftarrow h\left(\exp\left(\boldsymbol{\xi}^j_{k+1|k}\right)\bar{\chi}_{k+1|k}, \mathbf{n}^j_V\right)$;`// cubature points propagation`

$\delta\bar{\boldsymbol{\xi}}, \delta\bar{\mathbf{b}} \leftarrow \mathcal{QR}{}'(\mathbf{Y_0}, \mathbf{Y_N}, \mathbf{Y_j}, \boldsymbol{\xi}_j, \mathbf{N})$

$\bar{\chi} \leftarrow \exp\left(\delta\bar{\boldsymbol{\xi}}\right)\bar{\chi}, \bar{\mathbf{b}} \leftarrow \bar{\mathbf{b}} + \delta\bar{\mathbf{b}}$,

output: $\bar{\chi}_{k+1}$, $\mathbf{K}_{k+1}$,$\mathbf{P}_{k+1}$.

is used to transfer the state variables and variance durin the whole process, avoiding the complicated Jacobian solvin process of the IMU motion equations and the observatio equations, while, the matrix Lie group performs cubature poir sampling, which extends the cubature transformation metho to manifold space. The state vector is represented by a high dimensional Lie group matrix, which has a explicit structur than the hybrid representation.

V. Simulation Results

A. Illustrative High-Dimensional Example

We consider a general form of the multi-quadric function,

$$y = \left(\sqrt{1 + \mathbf{x}^T\mathbf{x}}\right)^p, \tag{43}$$

where $\mathbf{x}$ is an n-dimensional Gaussian random variable with mean μ, and covariance Σ, while p takes values $p = 1, -1, -3, -5$. Our objective with this example is to use different methods in order to compute the first two order (uncentralized) moments with different dimensions and prior information, namely (i) the UKF-LG, (ii) the InEKF, and (iii) the Right-CKF-LG,

$$\begin{aligned} \mathbb{E}(y) &= \int_{\mathbb{R}^n}\left(\sqrt{1 + \mathbf{x}^T\mathbf{x}}\right)^p \mathcal{N}(\mathbf{x}; \mu, \Sigma)d\mathbf{x}, \\ \mathbb{E}\left(y^2\right) &= \int_{\mathbb{R}^n}\left(1 + \mathbf{x}^T\mathbf{x}\right)^p \mathcal{N}(\mathbf{x}; \mu, \Sigma)d\mathbf{x} \end{aligned} \tag{44}$$

The Monte Carlo method with 10,000 samples is used to obtain the optimal estimate. We use the Kullback Leibler (KL) divergence to evaluate the filter estimated statistics against the optimal statistics. μ is set to be zero, and covariance matrices are randomly generated with diagonal entries being all equal to σ_0^2 for 50 independent runs.

Fig.1 plot the KL divergence of the InEKF and Right-CKF-LG estimations, respectively, when the dimension n increases for $\sigma_0^2 = 1$. However, since UKF-computed covariance matrix is not guaranteed to be positive with increasing dimension, the unavailability of the square-root covariance causes the UKF to halt, and thus, we have not presented the results for UKF-LG. It is evident from Fig. 1, both InEKF and Right-CKF-LG estimations degrade when the dimension increases. When p changes from 1 to -5, we can also notice the degradation as expected. CKF uses the spherical-radial cubature rule to numerically compute the moment integrals in the form non-linear function. The integrand is obtained from the product of a multi-quadric function and the standard Gaussian density over the radial variable. It has a peak occurring at a distance from the origin proportional to $\sqrt{n}$. On the other hand, the cubature points are located at a radius that scales with $\sqrt{n}$. Note that the third-degree spherical-radial rule is exact for polynomials of all odd degrees even beyond three. These may be the reasons for the increased accuracy obtained using the Right-CKF-LG. Furthermore, for all p being considered, the InEKF estimation is inferior to that of Right-CKF-LG.

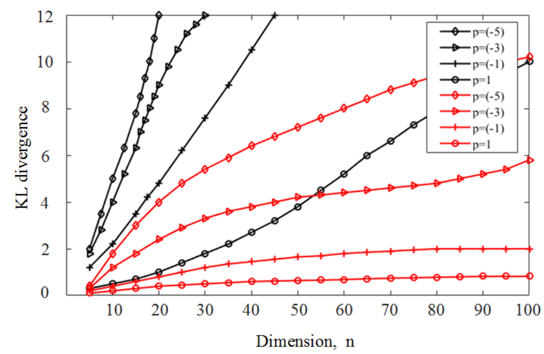

Fig. 1. Variation of KL divergence as dimension n increases, black line-InEKF, red line-Right-CKF-LG.

B. Effect of Noise on Estimates

We further explore the influences of estimation errors to Right-CKF-LG when the Gaussian nature of measurement noises is violated. For such a purpose, we let the measurement noise follow a Gaussian mixture as below,

$$\mathbf{n}_k \sim 0.5\mathcal{N}(\mathbf{0}, R_1) + 0.5\mathcal{N}(\mathbf{0}, R_2) \tag{45}$$

We employ the root-mean square error (RMSE) of position, orientation, and velocity to evaluate the nonlinear filter performances. The RMSE generates a combined measure of the bias and variance of the filter estimation. Since UKF halts when the state dimension is high, we have not presented the UKF results here. Not surprisingly, as can be seen from Figs. 2(a), 2(b) and 2(c), both filters diverge due to a mismatch between the filter-design assumption and the non-Gaussian nature. It can also be perceived that during the divergence, the true RMSEs of both filters exceed their corresponding estimated values. Although there are some differences between covariance estimations for the two filters, it's worth noting that unlike InEKF, Right-CKF-LG improves the performance after a short period of divergence, and it does not allow estimation errors to accumulate continuously, thereby avoiding an immediate 'blow-up'.

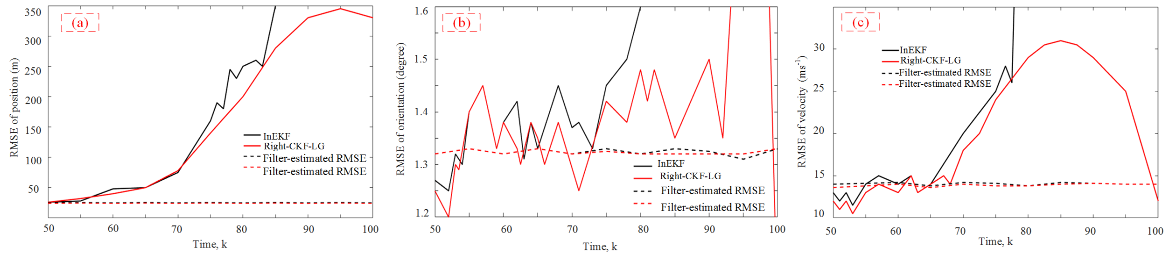

Fig. 2. Effects of noise distribution on estimations. (a) RMSE in position, true RMSE (black solid line-InEKF, red solid line-Right-CKF-LG), filter-estimated RMSE (black dashed line-InEKF, red dashed line-Right-CKF-LG) . (b) RMSE in orientation. (c) RMSE in velocity.

C. The advantage of using Lie groups to represent uncertainty in consistency and accuracy

To verify the theoretical contributions of this paper, 50 Monte Carlo simulations capitalized to juxtapose SRCKF with Right-CKF-LG, considering a robot equipped with an IMU and a camera for sufficient 6-DOF motion. In a simulated environment, the camera can observe landmarks with sufficient overlap between consecutive frames. For robust estimation, we only use landmarks for the update step if they are captured more than 5 times by cameras in the current range. In each round of Monte Carlo simulation, the initial estimation is set to the ground truth. The IMU and camera measurements are based on random noise generated by the same trajectory. The simulation results are shown in Fig. 3 and Fig. 4.

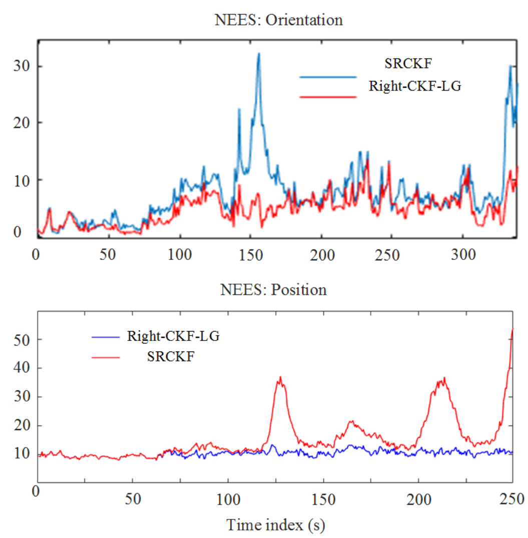

Fig. 3. NEES of Monte Carlo simulations.

The root mean squared error (RMSE) and normalized estimation error squared (NEES) are used to indicate the estimation performance. RMSE evaluates accuracy while NEES evaluates consistency. Since Right-CKF-LG is invariant to random translations and rotations around the direction of gravity, it prevents unintended information gain compared to SRCKF, therefore, Right-CKF-LG significantly outperforms SRCKF, especially in terms of consistency.

For each condition under different noise level, 100 Monte Carlo simulations are performed. For every simulation, the landmarks are incrementally added into the state vector while

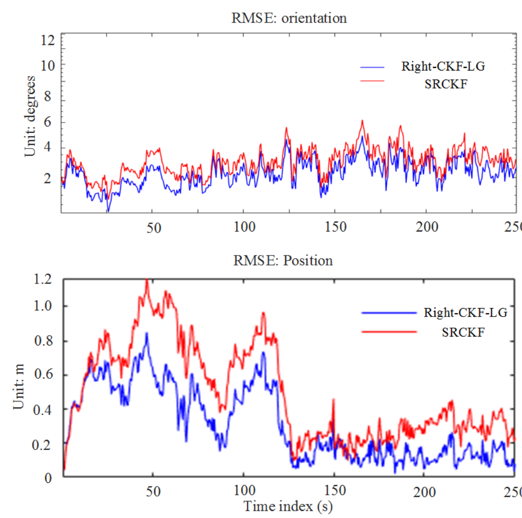

Fig. 4. RMSE of Monte Carlo simulations.

TABLE I
CONSISTENCY UNDER DIFFERENT NOISE LEVELS

$\delta_d = 1\%, \delta_b = 1\%$	Right-CKF-LG	SRCKF	UKF-LG	InEKF	$SE(3)$-CKF
RMSE of position (m)	0.25	0.45	0.29	0.32	0.65
RMSE of orientation (degree)	0.0062	0.0072	0.0071	0.0066	0.0083
NEES of pose	1.09	10	1.13	1.37	1.70
$\delta_d = 5\%, \delta_b = 5\%$	Right-CKF-LG	SRCKF	UKF-LG	InEKF	$SE(3)$-CKF
RMSE of position (m)	1.20	3.97	1.33	2.20	3.90
RMSE of orientation (degree)	0.029	0.044	0.031	0.045	0.041
NEES of pose	1.09	92	1.14	3.30	3.10

the initial covariance of robot is set to be zero matrix. The simulation results are summarized in Table I, where σ_d and σ_b are the odometry and observation noise levels, respectively, satisfying $\mathbf{Q}_n = \sigma_d^2 \operatorname{diag}\left(|\mathbf{u}_{n,1}|^2, \ldots, |\mathbf{u}_{n,k}|^2\right)$ and $\mathbf{N}_n^i = \sigma_d^2 \operatorname{diag}\left(\left|\mathbf{Z}_{n,1}^i\right|^2, \ldots, \left|\mathbf{Z}_{n,k}^i\right|^2\right)$ with $\mathbf{Z}_n^i$ being the ground truth of the coordinates of landmark.

As shown in Table I, the estimations of SRCKF and $SE(3)$-CKF are less effective in terms of consistency even under low noise level ($\delta_d = 1\%, \delta_b = 1\%$). This response is to be expected since both methods have no invariance property to deterministic rigid body transformation or stochastic identity transformation. Alternatively, when under high noise levels ($\delta_d = 5\%, \delta_b = 5\%$), both UKF-LG and InEKF do not perform optimally and subjected to larger linearization errors as compared with Right-CKF-LG in their propagation steps due to the coordinate transformation applied to the landmarks.

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2022.3214293

VI. Experimental results

This section verifies the performance of the CKF represented by the high-dimensional matrix Lie group in VIO. In our study, we have exploited benchmark public EuRoc MAV database. To compute the robot orientation and position errors, we have compared analyzed algorithms against UKF-LG algorithm, $SE(3)$-UKF algorithm [21], and InEKF. The Several algorithms use the filtering method to realize VIO, and all use the Lie group method to represent the motion.

A. Performance comparison of various algorithms under the same number of observed features

Fig. 5 describes the running results of Right-CKF-LG with EuRoc dataset. The black line represents the real trajectory, while the blue line represents the trajectory estimated by the proposed algorithm. As shown in the figure, the estimated trajectory and the real trajectory basically coincide for position error diversification. However, since the visual odometry has no closed-loop detection, the position error accumulates continuously, and the movement changes rapidly at the sharp turn of the robot, and the error accumulation is large.

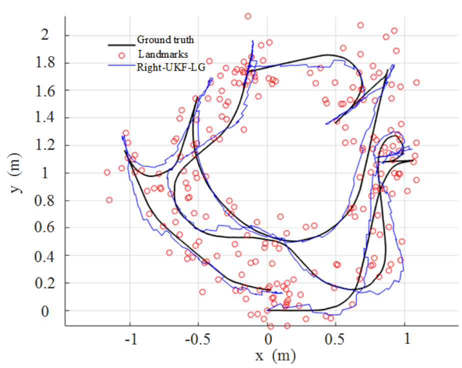

Fig. 5. The estimated trajectories from Right-CKF-LG.

Fig. 6(a) and 6(b) show the average error between the orientation, the estimated position and the real trajectory. Initially, the orientation error is similar for all four algorithms, then it increases at length, the growth rate and fluctuation range of the proposed Right-CKF-LG are significantly lower than the others. Similarly, as the running time increases, the cumulative error of the estimated position amplifies. The average error of the Right-CKF-LG algorithm starts to be slightly lower than the other three algorithms after 17s of operation, which is evident in Fig. 6(a). After running for 35s, the average error is significantly lower than the other three algorithms, as shown in Fig. 6(b). The results show that the proposed algorithm is effective in improving the performance of pose estimation.

Fig. 7 shows the velocity estimates, which again proves that the integrated system improves the localization accuracy. Right-CKF-LG is invariant to stochastic rigid body transformation, and thus, performs the best among opted algorithms. Fig. 8 explicitly indicates that, Right-CKF-LG also has consistency properties in terms of the 95% confidence bound.

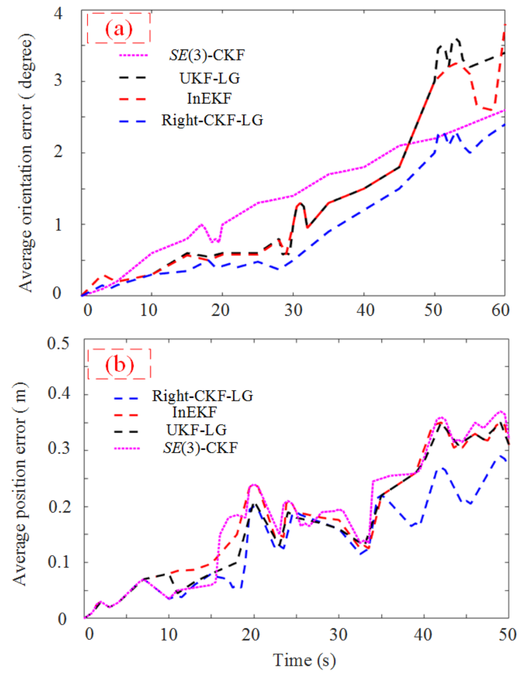

Fig. 6. Comparison of pose errors of different algorithms, (a) the orientation, (b) the estimated position.

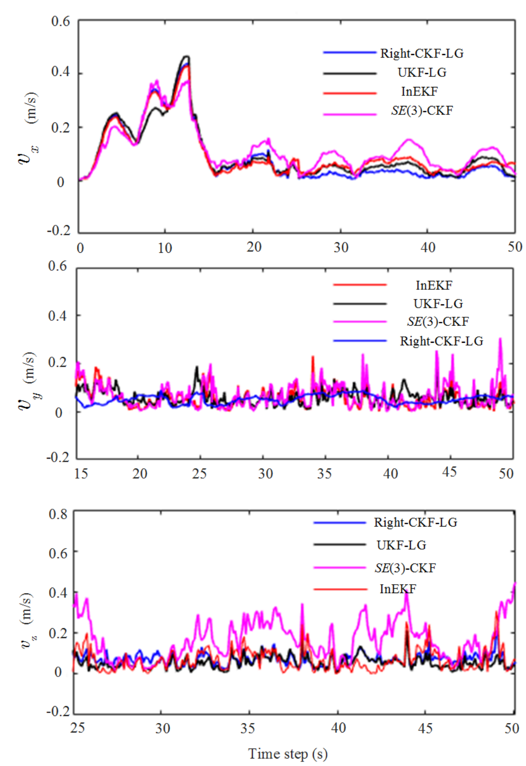

Fig. 7. Comparison of estimated velocities. (a) is the x-direction velocity. (b) is the y velocity.(c) is the z velocity.

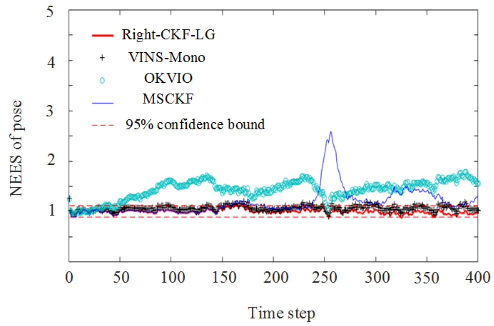

Fig. 8. Comparison of pose estimation in terms of consistency.

TABLE II
TIME-CONSUMING IN STATE UPDATE (UNIT: S)

Algorithms	Number of observation features				
	20	25	30	35	40
UKF-LG	7.6542	10.1269	13.1574	18.7296	23.3143
SE(3)-UKF	7.7998	10.1133	13.8469	17.7949	22.9073
InEKF	5.3353	8.4114	11.6172	13.6051	17.7390
Right-CKF-LG	7.6021	10.0811	13.0762	18.4982	23.2170

B. Influences of different observation features on state estimation

The dimension of the state variable in the Right-CKF-LG is $3m + 15$, with m being the number of observed features, while the number of sampling cubature points is twice that of the dimension of the augmented state variable. Therefore, the number of tracking feature points directly affects the algorithm operation time, while the transmitted observation noise impacts on the VIO.

We have conducted several experiments to verify the impacts of observation features on algorithm performances, including the estimation accuracy of position, orientation, and the average time-consuming of filtering state update. The RMSE of the entire pose trajectory, the update time of different algorithms are evaluated when the number of observed features is set to be 20, 25, 30, 35 and 40, respectively.

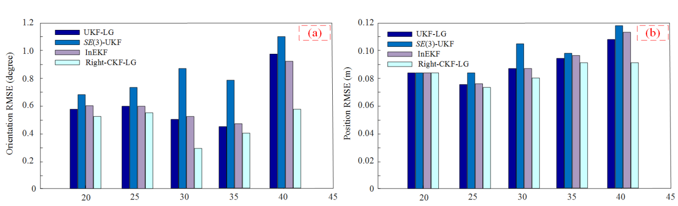

Fig. 9. RMSE of different observation features, (a) Orientation RMSE , (b) Position RMSE.

Results in Fig.9 show that Right-CKF-LG performs better than the others in terms of pose estimation accuracy for different observed features.

It can be seen from Table II that the time-consumption of the four algorithms increases with increment in tracking feature points. In aforementioned algorithms, InEKF takes the least time with various observed features, while both UKFs and CKFs require more computational power under different cases. The propagation requires much more computational power than the update for CKFs solutions, since the IMU (propagation) frequency is much larger than the camera (update) frequency. However, the estimation performance of InEKF is not optimal.

For orientation estimation, the valued performance of Right-CKF-LG is similar to when the feature numbers are 30 and 35, yet when the observed feature number is 35, the time-consumption is increased by 51% in the updated step. Whereas for position estimation, the position RMSE error is relatively small when the observed features are 25 and 30, however, the time consumption of the state update step is increased by 24%. From such results, we observe that too many or too few observed features will degrade the estimation performances, and thus, a reasonable number of observation features should be selected to balance the estimation accuracy and the time-consuming. In this study, we opted the feature number to be 30.

Based on the UKF theory on Lie groups, the invariant CKF has the advantage of featuring a complete Lie group structure. With a matrix Lie group representing the rotation of a rigid body, the quaternion representation method over parameters and normalization constraints can be avoided. In addition, the proposed filter can be easily adapted to small modifications in the model through parameters or other sensors. Furthermore, it is necessary to fuse more sensory information to improve the positioning accuracy and robustness.

VII. CONCLUSION

This paper studies accurate robot state estimation in VIO system. To reduce system internalization errors and improve the odometry robustness, an invariant CKF scheme combining CKF and invariant filtering theory was proposed. By utilizing the Lie group cubature transformation to extend CKF from Euclidean space to the manifold space, the uncertainty of state variable was transferred to avoid over-parameterization and normalization constraints, while the filter instability caused by growth in UKF state dimension was alleviated. Monte Carlo simulations were conducted with a public dataset to compare the proposed algorithm with the other ones for robot state estimations. Acquired results demonstrated that the scheme invariance largely impacts on the estimator consistency and accuracy, and the proposed Right-CKF-LG scheme outperforms SRCKF, UKF-LG, InEKF, SE(3)-CKF for robot pose estimation under different cases.

ACKNOWLEDGMENT

This study is supported by the Key Research and Development Program of Shaanxi, China (2021SF-342), China Postdoctoral Science Foundation (2018M641013), Postdoctoral Science Foundation of Shaanxi Province, China (2018BSHY-DZZ05), National Natural Science Foundation of China (Grant No. 51275423), the Science and Technology program of Gansu Province (20JR5RA483).

Professor Xiaojun Yu* will be the Corresponding author of this paper.

REFERENCES

[1] Sun K, Mohta K, Plrommer B, et al, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Letter,* 2018,3(2):965-972.

[2] Habiba Batti,Chiraz Ben Jabeur,Hassene Seddik. "Autonomous smart robot for path predicting and finding in maze based on fuzzy and neuro-Fuzzy approaches." *ASIAN JOURNAL OF CONTROL,*2020,23(1):3-12.

[3] Zhao S, Lin F, Peng K, et al. "Vision-aided Estimation of Attitude, Velocity, and Inertial Measurement Bias for UAV Stabilization."*Journal of Intelligent and Robotic Systems,* 2016, 81(3-4):531-549.

[4] Santoso F, Garratt M A, Anavati S G. "Visual-inertial navigation systems for aerial robotics: Sensor fusion and technology," *IEEE Transactions on Automation Science and Engineering,* vol. 14, No. 1, pp. 260-275, 2017.

[5] Paul M K, Wu K J, Hesch J A, et al. "A comparative analysis of tightly-coupled monocular, binocular, and stereo VINS," *IEEE International Conference on Robotics and Automation (ICRA),* Singapore: IEEE,2017: 260-275.

[6] V, Engel J, Stueckler J, et al. " visual-inertial odometry with stereo cameras.," *IEEE International Conference on Robotics and Automation (ICRA),*Stochholm: IEEE, 2016: 1885-1892.

[7] Engel J, Stückler J,"Large-scale direct SLAM with stereo cameras,"*IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),*Hamburg: IEEE, 2015: 1935-1942.

[8] Bloesch M, Omari S, Hutter M, et al. "Robust visual inertial odometry using a direct EKF-based approach,"*IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Hamburg: IEEE, 2015: 298-304.

[9] G.P.Huang,A.I.Mourikis,and S.I.Roumeliotis, "A quadratic complexity observability-constrained unscented Kalman filter for SLAM," *IEEE Transactions on Robotics,* vol. 29, no. 5, pp. 1226-1243,2013.

[10] Ienkaran Arasaratnam, Simon Haykin. "Cubature Kalman Filtering for Continuous-Discrete Systems: Theory and Simulations," *IEEE TRANSACTIONS ON SIGNAL PROCESSING,* VOL. 58, NO. 10, 2010.

[11] Forster C, Carlone L, Dellaert F, et al. " On-manifold preintegration for real-time visual-inertial odometry," *IEEE Transactions on Robotics,* 2016, 33(1): 1-21.

[12] Heo S, Chan G P. "Consistent EKF-based visual-inertial odometry on matrix Lie group,"*IEEE Sensors Journal,*2018, 18(9): 3780-3788.

[13] Brossard M, Bonnabel S, Barrau A. "Invariant Kalman filtering for visual inertial SLAM,"*The 21st International Conference on Information Fusion (FUSION),* Cambridge: IEEE, 2018: 2021-2028.

[14] G. Chirikjian, "Stochastic Models, Information Theory, and Lie Groups,Volume 2: Analytic Methods and Modern Applications," *Applied and Numerical Harmonic Analysis,* Birkh¨auser Boston, 2011.

[15] T. D. Barfoot, "State Estimation for Robotics," *Cambridge University Press,*2017.

[16] Barrau A, Bonnabel S. "The invariant extended Kalman filter as a stable observer," *IEEE Transactions on Automatic Control,* vol. 62, No. 4, pp. 1797-1812, 2017.

[17] Barrau A, Bonnabel S. "An EKF-SLAM algorithm with consistency properties,"*https://arxiv.org.labs/1510.06263,*[2016-09-02].

[18] Zhang T, Wu K, Song J, et al. "Convergence and consistency analysis for a 3D invariant-EKF SLAM,"*IEEE Robotics Automation Letters,* 2017, 2(2): 733-740.

[19] Zhang T, Wu K, Su D, et al. " An invariant-EKF VINS algorithm for improving consistency," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Vancouver: IEEE: 2017: 1578-1585.

[20] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. Achtelik, and R. Siegwart, "The EuRoC MAV datasets," The International Journal of Robotics Research, 2015.

[21] G. Loianno, M. Watterson, and V. Kumar, "Visual inertial odometry for quadrotors on SE(3)," in IEEE International Conference On Robotics and Automation (ICRA) , pp. 1544–1551, IEEE, 2016.