Lie Algebraic Unscented Kalman Filter for Pose Estimation

Alexander M. Sjøberg, Olav Egeland

***Abstract*—An unscented Kalman filter for matrix Lie groups is proposed where the time propagation of the state is formulated on the Lie algebra. This is done with the kinematic differential equation of the logarithm, where the inverse of the right Jacobian is used. The sigma points can then be expressed as logarithms in vector form, and time propagation of the sigma points and the computation of the mean and the covariance can be done on the Lie algebra. The resulting formulation is to a large extent based on logarithms in vector form and is, therefore, closer to the UKF for systems in $\mathbb{R}^n$. This gives an elegant and well-structured formulation which provides additional insight into the problem, and which is computationally efficient. The proposed method is in particular formulated and investigated on the matrix Lie group $SE(3)$. A discussion on right and left Jacobians is included, and a novel closed form solution for the inverse of the right Jacobian on $SE(3)$ is derived, which gives a compact representation involving fewer matrix operations. The proposed method is validated in simulations.**

***Index Terms*—Matrix Lie Group, Unscented Kalman Filter**

I. Introduction

The use of Lie group theory for attitude and pose estimation has received considerable attention in the research literature. The reason for this is that the set of rotation matrices $SO(3)$ and the set of homogeneous transformation matrices $SE(3)$ are both matrix Lie groups. Matrix Lie groups have a number of properties that are useful in the design of estimators and observers. In addition, unit quaternions form a Lie group, and some of the design techniques for quaternion estimators and observers can be related to their Lie group properties. The main branches of methods for Lie group estimators are based on Kalman filtering and nonlinear observer design.

Early work on nonlinear attitude estimation and control with quaternions is found in [40], [48] and [19], where the structural properties of the unit quaternions were used in the design. This was used in navigation for pose estimation in [46] and for attitude estimation in [45], where bias estimation was included. An important development in Kalman filtering based on quaternions was the multiplicative extended Kalman filter [30] where the global attitude was represented by the 4-dimensional unit quaternion, while the 3-dimensional quaternion vector was estimated at each time step. This was later generalized to alternative 3-dimensional vector representations of attitude, including the modified Rodrigues parameters [16] and the rotation vector [38], which is the vector form of the logarithm in $SO(3)$ [37]. An unscented Kalman filter (UKF) [26] was developed for attitude estimation in [15], where the kinematic differential equation for the quaternions was used for the time propagation of the sigma points. Another example of attitude estimation on quaternions using the UKF is found in [41]. The multiplicative extended Kalman filter for quaternions have been extended to pose estimation by introducing dual quaternions [20], while in [17] a UKF was developed for pose estimation using dual modified Rodrigues parameters.

An important development was the nonlinear complementary filter [34], which was an attitude observer where the global attitude was represented by a rotation matrix, and the observation error was represented by the 3-dimensional vector form of the anti-symmetric part of the rotation matrix. The resulting filter is robust and well suited for low-cost sensors, and for a number of different sensor configurations. The nonlinear complementary filter was generalized to the special linear group in [33], and to $SE(3)$ in [1]. A related work on pose estimation is found in [39], where vision and inertial sensors are used. A nonlinear observer for Lie groups based on Riemannian gradient descent was presented in [28], [24]. This work also addressed the invariance properties that were addressed in the symmetry-preserving observer of [5], [6], which was further developed to an invariant extended Kalman filter that was used as a stable observer on Lie groups [4], and for consistency in extended Kalman filtering for SLAM [9].

The concept of a concentrated Gaussian distribution on Lie groups was introduced in [47], [14], where a normal distribution on a Lie group was defined in terms of a normal distribution of the logarithm in vector form. This was further developed in [3], where this formulation was used for fusion of multiple measurements of pose. The formulation of [3] was used in [7] to formulate an extended Kalman filter (EKF) for matrix Lie groups where the covariance was calculated for the concentrated Gaussian distribution, and the time propagation of the state and the covariance was derived from the first order approximation of the Baker-Campbell-Hausdorff (BCH) formula. A similar approach was used in [12] for an extended information filter on matrix Lie groups. The method of [7], [8] was further developed in [42] where the time propagation of the state and the covariance of an EKF was computed on the Lie algebra using the kinematic differential equations of the logarithm.

In [22] a UKF was formulated for Riemannian manifolds. This was done by generating sigma points as elements of the manifold, and then calculating the mean by minimization on the manifold, while the covariance was calculated in the tangent plane of the mean. It was remarked that the sigma points and the mean can alternatively be calculated in the tangent plane. In [23] a UKF framework for sensor fusion

The authors are with the Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. e-mail: alexander.m.sjoberg@ntnu.no, olav.egeland@ntnu.no.

on manifolds was presented where the sigma points were given on the manifold. A UKF for quadrotors on $SE(3)$ was presented in [31], where the sigma points were computed on the manifold. In [10] the concept of concentrated Gaussian distributions was used to formulate a UKF for Lie groups where the sigma points are in the Lie algebra, while the time propagation is formulated on the Lie group. In [36] a systematic overview of Riemannian extensions of UKFs was presented based on the formalism in [35]. A simulation study was included where a UKF was implemented for unit quaternions, where the sigma points were computed on the manifold, and the mean was found as an optimization problem on the manifold. The paper stated that future work should focus on computationally efficient UKFs for Riemannian manifolds for real-time applications. In [32] a UKF is formulated where the sigma points are calculated in the Lie algebra, while the time propagation was in the manifold.

The main contribution of the present paper is that the time propagation is formulated in terms of the kinematic differential equation of the logarithm, using the inverse of the right Jacobian. Then time propagation of the sigma points can be formulated on the Lie algebra, and moreover, the mean and the covariance can be computed on the Lie algebra. The covariance matrix is transformed between different tangent planes based on the BCH formula using the right Jacobian. The measurement update is based on [10]. The resulting formulation is to a large extent given in terms of logarithms in vector form, which makes the proposed UKF more similar to the original formulation for $\mathbb{R}^n$. This may lead to added insight and ease of implementation. Moreover, some of the steps of the method will be computationally efficient. In particular, time propagation of the group element and optimization on the manifold is avoided, and the method involves few calculations of exponentials and logarithms. The right Jacobian is important in our method as it appears in the kinematic differential equation of the logarithm. A new closed form solution for the inverse right and left Jacobian in $SE(3)$ is derived in the paper based on [11], [3].

The paper is organized as follows. Section 2 presents basic theory on Lie groups and Lie algebras including probability distributions and the left and right Jacobians. In Section 3 a new closed form solution for the inverse right and left Jacobian in $SE(3)$ is derived. Then in Section 4 a Lie Algebraic UKF on $SE(3)$ is presented. Finally, the performance of the proposed UKF is demonstrated in simulations.

II. Preliminaries

A. Matrix Lie groups

Consider the matrix Lie group G and the associated Lie algebra $\mathfrak{g}$ [14], [21]. The exponential of $\mathfrak{u} \in \mathfrak{g}$ is $\boldsymbol{X} = \exp \mathfrak{u} = \sum_{k=0}^{\infty} \mathfrak{u}^k/k! \in G$. Then $\mathfrak{u} = \log \boldsymbol{X}$ is the logarithm of $\boldsymbol{X}$ locally around the identity $\mathrm{id} \in G$. Let $\mathfrak{a}, \mathfrak{b} \in \mathfrak{g}$ be elements of the Lie algebra. Then the adjoint map $ad(\mathfrak{a})$ is given by $ad(\mathfrak{a})\mathfrak{b} = [\mathfrak{a}, \mathfrak{b}] = \mathfrak{a}\mathfrak{b} - \mathfrak{b}\mathfrak{a}$, which is the Lie bracket.

The time derivative of the exponential function is

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\left(\exp \mathfrak{u}(t)\right) = \left(T_{\mathfrak{u}} \exp\right) \dot{\mathfrak{u}}(t) \tag{1}$$

where $T_{\mathfrak{u}} \exp : \mathfrak{g} \to T_{\exp \mathfrak{u}} G$ is the linear mapping [18]

$$\begin{aligned} T_{\mathfrak{u}} \exp &= T_{\mathrm{id}} R(\exp \mathfrak{u}) \circ \sum_{k=0}^{\infty} \frac{(+ad(\mathfrak{u}))^k}{(k+1)!} \\ &= T_{\mathrm{id}} L(\exp \mathfrak{u}) \circ \sum_{k=0}^{\infty} \frac{(-ad(\mathfrak{u}))^k}{(k+1)!} \end{aligned} \tag{2}$$

where $T_{\mathrm{id}} R(\boldsymbol{X})$ and $T_{\mathrm{id}} L(\boldsymbol{X})$ are the tangent mappings at the identity of the right and left translation by $\boldsymbol{X}$, respectively. In the case of a matrix group, $T_{\mathrm{id}} R(\boldsymbol{X})\mathfrak{a} = \mathfrak{a}\boldsymbol{X}$ and $T_{\mathrm{id}} L(\boldsymbol{X})\mathfrak{a} = \boldsymbol{X}\mathfrak{a}$. Let $\boldsymbol{a} = [a_1, \ldots, a_n]^{\mathrm{T}} \in \mathbb{R}^n$ be the vector representation of $\mathfrak{a} \in \mathfrak{g}$. The notation $[\mathfrak{a}]_G^{\vee} = \boldsymbol{a}$ and $[\boldsymbol{a}]_G^{\wedge} = \mathfrak{a}$ is used [7]. Moreover, $[\mathbf{ad}(\boldsymbol{a})\boldsymbol{b}]_G^{\wedge} = ad(\mathfrak{a})\mathfrak{b}$ where $[\boldsymbol{b}]_G^{\wedge} = \mathfrak{b}$. The time derivative of the exponential is then given by

$$\dot{\boldsymbol{X}} = [\boldsymbol{J}_l(\mathbf{ad}(\boldsymbol{u}))\dot{\boldsymbol{u}}]_G^{\wedge}\boldsymbol{X} = \boldsymbol{X}[\boldsymbol{J}_r(\mathbf{ad}(\boldsymbol{u}))\dot{\boldsymbol{u}}]_G^{\wedge} \tag{3}$$

where

$$\boldsymbol{J}_l(\mathbf{ad}(\boldsymbol{u})) = \sum_{k=0}^{\infty} \frac{(\mathbf{ad}(\boldsymbol{u}))^k}{(k+1)!} \tag{4}$$

is the left Jacobian and $\boldsymbol{J}_r(\mathbf{ad}(\boldsymbol{u})) = \boldsymbol{J}_l(-\mathbf{ad}(\boldsymbol{u}))$ is the right Jacobian. It is noted that in view of the Cayley-Hamilton theorem, the Jacobian can be evaluated from a finite number of terms of the form $(\mathbf{ad}(\boldsymbol{u}))^k$.

The kinematic differential equation for $\boldsymbol{X} \in G$ is given by

$$\dot{\boldsymbol{X}} = [\boldsymbol{v}_l]_G^{\wedge}\boldsymbol{X} = \boldsymbol{X}[\boldsymbol{v}_r]_G^{\wedge} \tag{5}$$

where $\boldsymbol{v}_l, \boldsymbol{v}_r \in \mathbb{R}^n$ are the right and left invariant velocities, respectively. Comparison with (3) leads to the kinematic differential equations [11]

$$\dot{\boldsymbol{u}} = \boldsymbol{J}_l^{-1}(\mathbf{ad}(\boldsymbol{u}))\boldsymbol{v}_l = \boldsymbol{J}_r^{-1}(\mathbf{ad}(\boldsymbol{u}))\boldsymbol{v}_r \tag{6}$$

The inverse of the left and right Jacobian is given by $\boldsymbol{J}_l^{-1}(\mathbf{ad}(\boldsymbol{u})) = \boldsymbol{J}_r^{-1}(-\mathbf{ad}(\boldsymbol{u})) = \sum_{k=0}^{\infty} \frac{B_k(\mathbf{ad}(\boldsymbol{u}))^k}{k!}$ where B_k are the Bernoulli numbers [3].

B. The Baker-Campbell-Hausdorff formula

Consider $\exp(\mathfrak{c}) = \exp(\mathfrak{a}) \exp(\mathfrak{b})$ where $\mathfrak{a}, \mathfrak{b}, \mathfrak{c} \in \mathfrak{g}$. A first order approximation in $\mathfrak{b}$ of the BCH formula is given in vector form as $\boldsymbol{c} = \boldsymbol{a} + \boldsymbol{J}_r^{-1}(\mathbf{ad}(\boldsymbol{a}))\boldsymbol{b}$ [27]. This leads to the two first order approximations

$$\exp([\boldsymbol{a}]_G^{\wedge}) \exp([\boldsymbol{b}]_G^{\wedge}) = \exp([\boldsymbol{a} + \boldsymbol{J}_r^{-1}(\mathbf{ad}(\boldsymbol{a}))\boldsymbol{b}]_G^{\wedge}) \tag{7}$$

$$\exp([\boldsymbol{a} + \boldsymbol{d}]_G^{\wedge}) = \exp([\boldsymbol{a}]_G^{\wedge}) \exp([\boldsymbol{J}_r(\mathbf{ad}(\boldsymbol{a}))\boldsymbol{d}]_G^{\wedge}) \tag{8}$$

which were used in [3], [8] and [12].

C. Gaussian Distributions on Matrix Lie Groups

It is noted that the normal distribution is of maximum entropy in $\mathbb{R}^n$, while this is not the case for Lie groups. Maximum entropy distributions for Lie groups are possible to find based on Souriau's work [2]. Alternative global approaches to find a probability density function (pdf) for $SO(3)$ are found in [29], and in [43] where a Bingham distribution was used for pose estimation. In this work our target has been to formulate a UKF, and therefore it is interesting to use a formulation based on a Gaussian distribution.

In the following we will restrict our discussion to the Lie groups $SO(3)$, $SE(2)$ and $SE(3)$. Consider the random variable $\boldsymbol{X} \in G$ and define the distance function $d(\boldsymbol{X}_1, \boldsymbol{X}_2) = \|(\log(\boldsymbol{X}_1^{-1}\boldsymbol{X}_2))_G^\vee\|$ for $\boldsymbol{X}_1, \boldsymbol{X}_2 \in G$. Then the pdf $p_X(\boldsymbol{X})$ of $\boldsymbol{X}$ is said to be highly concentrated around $\bar{\boldsymbol{X}}$ if $\int_{d(\bar{\boldsymbol{X}},\boldsymbol{X})<\epsilon} p_X(\boldsymbol{X})d\boldsymbol{X} \approx 1$ for $\epsilon \ll \pi$ [14, p. 381]. In this paper we will use $\boldsymbol{X} = \bar{\boldsymbol{X}}\exp([\boldsymbol{u}]_G^\wedge)$ and the pdf $p(\boldsymbol{u})$ of the logarithm, which gives the condition $\int_{\|\boldsymbol{u}\|<\epsilon} p(\boldsymbol{u})d\boldsymbol{u} \approx 1$ for $\epsilon \ll \pi$ for a highly concentrated distribution of the logarithm around zero. This was used for $SE(3)$ in [3], where where the Gaussian pdf $p(\boldsymbol{u}) \sim \mathcal{N}_{\mathbb{R}^n}(\boldsymbol{0}, \boldsymbol{P})$ of the logarithm was used, and the mean and covariance of the logarithm were given by

$$\boldsymbol{\mu} = \int_{\mathbb{R}^n} \boldsymbol{u}p(\boldsymbol{u})d\boldsymbol{u}, \quad \boldsymbol{\Sigma} = \int_{\mathbb{R}^n} \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}p(\boldsymbol{u})d\boldsymbol{u} \tag{9}$$

The pdf $p(\boldsymbol{u})$ will be highly concentrated around zero if the largest singular value of the covariance $\boldsymbol{\Sigma}$ is much smaller than π. Then the pdf $p(\boldsymbol{u})$ will be very small for $\|u\| \geq \pi$, and the normal distribution is a valid approximation with a folding or cutting technique [14], [3].

Next, consider a random variable $\boldsymbol{Y} = \exp([\boldsymbol{\xi}]_G^\wedge) \in G$, where $\boldsymbol{\xi} \sim \mathcal{N}_{\mathbb{R}^n}(\bar{\boldsymbol{\xi}}, \boldsymbol{Q})$ is normally distributed with nonzero mean. The zero-mean vector $\delta\boldsymbol{\xi} = \boldsymbol{\xi} - \bar{\boldsymbol{\xi}} \sim \mathcal{N}_{\mathbb{R}^n}(\boldsymbol{0}, \boldsymbol{Q})$ is introduced. Then from (8) it is seen that a first order approximation in $\delta\boldsymbol{\xi}$ is given by

$$\boldsymbol{Y} = \exp([\bar{\boldsymbol{\xi}} + \delta\boldsymbol{\xi}]_G^\wedge) = \bar{\boldsymbol{Y}}\exp([\boldsymbol{e}]_G^\wedge) \tag{10}$$

where $\bar{\boldsymbol{Y}} = \exp([\bar{\boldsymbol{\xi}}]_G^\wedge)$ and $\boldsymbol{e} = \boldsymbol{J}_r(\mathbf{ad}(\bar{\boldsymbol{\xi}}))\delta\boldsymbol{\xi} \sim \mathcal{N}_{\mathbb{R}^n}(\boldsymbol{0}, \boldsymbol{R})$ is normally distributed with zero mean and covariance

$$\boldsymbol{R} = \boldsymbol{J}_r(\mathbf{ad}(\bar{\boldsymbol{\xi}}))\boldsymbol{Q}\boldsymbol{J}_r^{\mathrm{T}}(\mathbf{ad}(\bar{\boldsymbol{\xi}})) \tag{11}$$

This result was derived in [7] for use in the update of an extended Kalman filter on a matrix Lie group. A related problem was treated in [13] where a merging algorithm for Gaussian components on G was developed.

D. Calculation of mean and covariance

Consider a set $\boldsymbol{Y}_i = \exp([\boldsymbol{\xi}_i]_G^\wedge) \in G$ of N Lie group elements with corresponding logarithms given in vector form by $\boldsymbol{\xi}_i$, where $\boldsymbol{\xi}_i$ is highly concentrated around zero. In [22] a UKF for Riemannian manifolds was proposed where the elements were expressed in terms of the mean $\boldsymbol{\mu} \in G$ as

$$\boldsymbol{Y}_i = \boldsymbol{\mu}\exp([\boldsymbol{\epsilon}_i]_G^\wedge) \tag{12}$$

where the mean was found from the minimization problem

$$\boldsymbol{\mu} = \arg\min_{\boldsymbol{\mu}\in G} d(\boldsymbol{Y}_i, \boldsymbol{\mu})^2 \tag{13}$$

for some distance function d. The empirical covariance was calculated from $\boldsymbol{P}_G = \frac{1}{N}\sum_{i=1}^N \boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_i^{\mathrm{T}}$ where $[\boldsymbol{\epsilon}_i]_G^\wedge = \log(\boldsymbol{\mu}^{-1}\boldsymbol{Y}_i)$. The calculation of the mean as the minimization problem (13) on G may be time consuming in real time applications.

Therefore, we propose a new formulation for a UKF on a Lie group where the mean of the set $\boldsymbol{Y}_i = \exp([\boldsymbol{\xi}_i]_G^\wedge)$ is calculated on the tangent space as

$$\bar{\boldsymbol{Y}} = \exp([\bar{\boldsymbol{\xi}}]_G^\wedge) \tag{14}$$

where

$$\bar{\boldsymbol{\xi}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{\xi}_i \tag{15}$$

It is seen from (10) that this leads to the first order approximation

$$\boldsymbol{Y}_i = \bar{\boldsymbol{Y}}\exp(\boldsymbol{e}_i) \tag{16}$$

where $\boldsymbol{e}_i = \boldsymbol{J}_r(\mathbf{ad}(\bar{\boldsymbol{\xi}}))\delta\boldsymbol{\xi}_i$, and $\delta\boldsymbol{\xi}_i = (\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}})$. The empirical covariance can then be calculated as

$$\boldsymbol{P} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{e}_i\boldsymbol{e}_i^{\mathrm{T}} = \boldsymbol{J}\left(\frac{1}{N}\sum_{i=1}^N \delta\boldsymbol{\xi}_i\delta\boldsymbol{\xi}_i^{\mathrm{T}}\right)\boldsymbol{J}^{\mathrm{T}} \tag{17}$$

where $\boldsymbol{J} = \boldsymbol{J}_r(\mathbf{ad}(\bar{\boldsymbol{\xi}}))$.

E. Calculation of mean by optimization

Consider the case where the distance function in (13) is $d(\boldsymbol{Y}_i, \boldsymbol{\mu})^2 = \boldsymbol{\epsilon}_i^{\mathrm{T}}\boldsymbol{\epsilon}_i$ which is the usual angular distance in $SO(3)$ and a left-invariant metric in $SE(3)$. Then the mean calculated by minimization on the group will be

$$\boldsymbol{\mu} = \arg\min_{\boldsymbol{\mu}\in G}([\log(\boldsymbol{\mu}^{-1}\boldsymbol{Y}_i)]_G^\vee)^{\mathrm{T}}[\log(\boldsymbol{\mu}^{-1}\boldsymbol{Y}_i)]_G^\vee \tag{18}$$

In comparison to this, for sufficiently small $\boldsymbol{\epsilon}_i$, the calculation of the mean $\bar{\boldsymbol{Y}}$ by calculating the average logarithm $[\bar{\boldsymbol{\xi}}]_G^\wedge$ in (15) corresponds to the minimization problem

$$\bar{\boldsymbol{Y}} = \arg\min_{\bar{\boldsymbol{Y}}\in G}([\log(\bar{\boldsymbol{Y}}^{-1}\boldsymbol{Y}_i)]_G^\vee)^{\mathrm{T}}\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}[\log(\bar{\boldsymbol{Y}}^{-1}\boldsymbol{Y}_i)]]_G^\vee \tag{19}$$

on G where $\boldsymbol{J} = \boldsymbol{J}_r(\mathbf{ad}(\bar{\boldsymbol{\xi}}))$. It is seen that the only difference between the two optimization problems (18) and (19) is the weighting matrix $\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$ in (19). For small $\bar{\boldsymbol{\xi}}$ it is seen from (4) that this weighting matrix will be close to the identity matrix, and it is reasonable to expect that $\bar{\boldsymbol{Y}}$ will be close to $\boldsymbol{\mu}$ for this distance function.

F. Time integration and discrete-time model

Euler's method for the kinematic differential equation (5) gives

$$\boldsymbol{X}(t_{k+1}) = \boldsymbol{X}(t_k)\exp(h\boldsymbol{v}_r(t_k)) \tag{20}$$

where $t_{k+1} = t_k + h$, h is the time step, and $\boldsymbol{v}_r$ is assumed to be constant over the time interval. Alternatively, Euler's method can be applied to (6), which gives

$$\boldsymbol{u}(t_{k+1}) = \boldsymbol{u}(t_k) + h\boldsymbol{J}_r^{-1}(\mathbf{ad}(\boldsymbol{u}(t_k)))\boldsymbol{v}_r(t_k) \tag{21}$$

From $\boldsymbol{X}(t_k) = \exp([\boldsymbol{u}(t_k)]_G^\wedge)$ and (7) it follows that

$$\boldsymbol{X}(t_{k+1}) = \exp([\boldsymbol{u}(t_k) + h\boldsymbol{J}_r^{-1}(\mathbf{ad}(\boldsymbol{u}(t_k)))\boldsymbol{v}_r(tk)]_G^\wedge) \tag{22}$$

is a first order approximation of (20). This means that the discretizations (20) and (21) give the same result to the first order. We will use the discretization (21) for our UKF.

Additional results on integration schemes based on (6) are found in [25] and [44].

III. Calculation of inverse Jacobian

In our proposed UKF for matrix Lie groups the inverse of the right Jacobian plays an important role. We will therefore take a closer look at expressions for the inverse of the right Jacobian, and a novel closed form solution for $SE(3)$ will be developed.

A. Jacobians in $SO(3)$

The logarithm in $SO(3)$ is the skew symmetric matrix $\hat{\boldsymbol{\theta}} \in so(3)$ corresponding to the vector $\boldsymbol{\theta}$, while $\mathbf{ad}(\boldsymbol{\theta}) = \hat{\boldsymbol{\theta}}$. The right Jacobian and its inverse are given in closed form as [11]

$$\boldsymbol{\Psi}_r(\hat{\boldsymbol{\theta}}) = \boldsymbol{I} - \frac{1-\cos\theta}{\theta^2}\hat{\boldsymbol{\theta}} + \frac{\theta-\sin\theta}{\theta^3}\hat{\boldsymbol{\theta}}^2 \tag{23}$$

$$\boldsymbol{\Psi}_r^{-1}(\hat{\boldsymbol{\theta}}) = \boldsymbol{I} + \frac{1}{2}\hat{\boldsymbol{\theta}} + \frac{1-\frac{\theta}{2}\cot\frac{\theta}{2}}{\theta^2}\hat{\boldsymbol{\theta}}^2 \tag{24}$$

where $\theta = \|\boldsymbol{\theta}\|$. The coefficients of the Jacobian and its inverse are well defined for all $\boldsymbol{\theta}$, which is verified by Taylor series expansion of the coefficients.

B. Jacobians in $SE(3)$

The logarithm of $\boldsymbol{T} = (\boldsymbol{R}, \boldsymbol{r}) \in SE(3)$ is given by

$$[\boldsymbol{\xi}]^\wedge_{SE(3)} = \begin{bmatrix} \hat{\boldsymbol{\theta}} & \boldsymbol{\rho} \\ \mathbf{0}^\mathrm{T} & 0 \end{bmatrix} \in se(3) \tag{25}$$

where $\hat{\boldsymbol{\theta}} \in so(3)$, $\boldsymbol{\rho} \in \mathbb{R}^3$ and $\boldsymbol{\xi} = [\boldsymbol{\theta}^\mathrm{T}, \boldsymbol{\rho}^\mathrm{T}]^\mathrm{T}$ is the vector form of the logarithm. The exponential map is [11], [37]

$$\boldsymbol{T} = \exp([\boldsymbol{\xi}]^\wedge_{SE(3)}) = \begin{bmatrix} \exp\hat{\boldsymbol{\theta}} & \boldsymbol{\Psi}_l(\hat{\boldsymbol{\theta}})\boldsymbol{\rho} \\ \mathbf{0}^\mathrm{T} & 1 \end{bmatrix} \tag{26}$$

where $\exp\hat{\boldsymbol{\theta}}$ is the exponential function in $SO(3)$, and $\boldsymbol{\Psi}_l(\hat{\boldsymbol{\theta}})$ is the left Jacobian in $SO(3)$. The logarithm can be computed from the skew symmetric matrix $\hat{\boldsymbol{y}} = \frac{1}{2}(\boldsymbol{R} - \boldsymbol{R}^\mathrm{T})$ as [25]

$$\hat{\boldsymbol{\theta}} = \frac{\sin^{-1}\|\boldsymbol{y}\|}{\|\boldsymbol{y}\|}\hat{\boldsymbol{y}}, \quad \boldsymbol{\rho} = \boldsymbol{\Psi}_l^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{r} \tag{27}$$

The matrix form of the adjoint map in $SE(3)$ is given by

$$\mathbf{ad}(\boldsymbol{\xi}) = \begin{bmatrix} \hat{\boldsymbol{\theta}} & \mathbf{0} \\ \hat{\boldsymbol{\rho}} & \hat{\boldsymbol{\theta}} \end{bmatrix} \tag{28}$$

The kinematic differential equation in terms of $\boldsymbol{T}$ is given by

$$\dot{\boldsymbol{T}} = [\boldsymbol{V}_l]^\wedge_{SE(3)}\boldsymbol{T} = \boldsymbol{T}[\boldsymbol{V}_r]^\wedge_{SE(3)} \tag{29}$$

where $[\boldsymbol{V}_l]^\wedge_{SE(3)} \in se(3)$ is the left velocity, and $[\boldsymbol{V}_r]^\wedge_{SE(3)} \in se(3)$ is the right velocity, which have vector forms

$$\boldsymbol{V}_r = \begin{bmatrix} \boldsymbol{\omega}_r \\ \boldsymbol{v}_r \end{bmatrix}, \quad \boldsymbol{V}_l = \begin{bmatrix} \boldsymbol{\omega}_l \\ \boldsymbol{v}_l \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}\boldsymbol{\omega}_r \\ \boldsymbol{R}\boldsymbol{v}_r + \hat{\boldsymbol{r}}\boldsymbol{R}\boldsymbol{\omega} \end{bmatrix} \tag{30}$$

It follows from (6) that the kinematic differential equation in terms of the logarithm is

$$\dot{\boldsymbol{\xi}} = \boldsymbol{\Phi}_l^{-1}(\mathbf{ad}(\boldsymbol{\xi}))\boldsymbol{V}_l = \boldsymbol{\Phi}_r^{-1}(\mathbf{ad}(\boldsymbol{\xi}))\boldsymbol{V}_r \tag{31}$$

where $\boldsymbol{\Phi}_l$ is the left Jacobian and $\boldsymbol{\Phi}_r$ is the right Jacobian.

The right Jacobian is given in closed form as [11]

$$\boldsymbol{\Phi}_r(\mathbf{ad}(\boldsymbol{\xi})) = \begin{bmatrix} \boldsymbol{\Psi}_r(\hat{\boldsymbol{\theta}}) & \mathbf{0} \\ \boldsymbol{Q}_r & \boldsymbol{\Psi}_r(\hat{\boldsymbol{\theta}}) \end{bmatrix} \tag{32}$$

where a closed form solution for the submatrix $\boldsymbol{Q}_r$ was found in [3]. Inversion of the matrix in (32) gives the expression

$$\boldsymbol{\Phi}_r^{-1}(\mathbf{ad}(\boldsymbol{\xi})) = \begin{bmatrix} \boldsymbol{\Psi}_r^{-1}(\hat{\boldsymbol{\theta}}) & \mathbf{0} \\ \boldsymbol{C}_r & \boldsymbol{\Psi}_r^{-1}(\hat{\boldsymbol{\theta}}) \end{bmatrix} \tag{33}$$

for the inverse of the right Jacobian, as presented in [11], where the submatrix $\boldsymbol{C}_r$ was unspecified. Later, the expression $\boldsymbol{C}_r = -\boldsymbol{\Psi}_r^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{Q}_r\boldsymbol{\Psi}_r^{-1}(\hat{\boldsymbol{\theta}})$ was obtained in [3]. It is noted that a closed form solution for $\boldsymbol{C}_r$ was not found, which means that a closed form solution for the inverse of the right Jacobian in $SE(3)$ has not been reported so far.

C. Closed form for inverse Jacobians in $SE(3)$

In this section we will derive a simple closed form solution for the inverse of the right and left Jacobians in $SE(3)$. In [11] it was shown that the inverse right Jacobian on $SE(3)$ can be computed as

$$\boldsymbol{\Phi}_r^{-1}(\mathbf{ad}(\boldsymbol{\xi})) = \boldsymbol{I} + \frac{1}{2}\mathbf{ad}(\boldsymbol{\xi}) + \gamma_1(\theta)\mathbf{ad}(\boldsymbol{\xi})^2 + \gamma_2(\theta)\mathbf{ad}(\boldsymbol{\xi})^4 \tag{34}$$

with $\gamma_1(y) = (4 - 3\alpha(y) - \beta(y))/(2y^2)$, $\gamma_2(y) = (2 - \alpha(y) - \beta(y))/(2y^4)$, $\alpha(y) = (y/2)\cot(y/2)$, $\beta(y) = (y/2)^2/\sin^2(y/2)$. This was used in [11] to derive (33).

It follows from (28) that

$$\mathbf{ad}(\boldsymbol{\xi})^k = \begin{bmatrix} \hat{\boldsymbol{\theta}}^k & \mathbf{0} \\ \boldsymbol{s}_k & \hat{\boldsymbol{\theta}}^k \end{bmatrix} \tag{35}$$

where $\boldsymbol{s}_k = \sum_{i=0}^{k-1}\hat{\boldsymbol{\theta}}^{k-i-1}\hat{\boldsymbol{\rho}}\hat{\boldsymbol{\theta}}^i$. From this in combination with (33), (34) and (35), the matrix $\boldsymbol{C}_r$ must be of the form

$$\boldsymbol{C}_r = \frac{1}{2}\boldsymbol{s}_1 + \gamma_1(\theta)\boldsymbol{s}_2 + \gamma_2(\theta)\boldsymbol{s}_4 \tag{36}$$

where $\boldsymbol{s}_1 = \hat{\boldsymbol{\rho}}$, $\boldsymbol{s}_2 = \hat{\boldsymbol{\theta}}\hat{\boldsymbol{\rho}} + \hat{\boldsymbol{\rho}}\hat{\boldsymbol{\theta}}$ and $\boldsymbol{s}_4 = \hat{\boldsymbol{\theta}}^3\hat{\boldsymbol{\rho}} + \hat{\boldsymbol{\theta}}^2\hat{\boldsymbol{\rho}}\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\theta}}\hat{\boldsymbol{\rho}}\hat{\boldsymbol{\theta}}^2 + \hat{\boldsymbol{\rho}}\hat{\boldsymbol{\theta}}^3$. The expression for $\boldsymbol{s}_4$ is simplified by observing that $\hat{\boldsymbol{\theta}}^2\hat{\boldsymbol{\rho}}\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\theta}}\hat{\boldsymbol{\rho}}\hat{\boldsymbol{\theta}}^2 = \hat{\boldsymbol{\theta}}(\hat{\boldsymbol{\rho}}\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\theta}}\hat{\boldsymbol{\rho}})\hat{\boldsymbol{\theta}} = -2(\boldsymbol{\theta}^\mathrm{T}\boldsymbol{\rho})\hat{\boldsymbol{\theta}}^2$. This in combination with $\hat{\boldsymbol{\theta}}^3 = -\theta^2\hat{\boldsymbol{\theta}}$ gives

$$\boldsymbol{s}_4 = -\theta^2\boldsymbol{s}_2 - 2(\boldsymbol{\theta}^\mathrm{T}\boldsymbol{\rho})\hat{\boldsymbol{\theta}}^2 \tag{37}$$

Insertion of this expression for $\boldsymbol{s}_4$ in (36) along with the expressions for $\gamma_1(\theta)$ and $\gamma_2(\theta)$ gives the closed form solution

$$\begin{aligned}\boldsymbol{C}_r = \frac{1}{2}\hat{\boldsymbol{\rho}} &+ \frac{1-\alpha(\theta)}{\theta^2}(\hat{\boldsymbol{\theta}}\hat{\boldsymbol{\rho}} + \hat{\boldsymbol{\rho}}\hat{\boldsymbol{\theta}}) \\ &+ \frac{\alpha(\theta)+\beta(\theta)-2}{\theta^4}(\boldsymbol{\theta}^\mathrm{T}\boldsymbol{\rho})\hat{\boldsymbol{\theta}}^2\end{aligned} \tag{38}$$

This gives the desired closed form solution for the inverse of the right Jacobian by inserting (24) and (38) into (33). The closed form solution for the inverse of the left Jacobian is then

$$\boldsymbol{\Phi}_l^{-1}(\mathbf{ad}(\boldsymbol{\xi})) = \boldsymbol{\Phi}_r^{-1}(-\mathbf{ad}(\boldsymbol{\xi})) = \begin{bmatrix} \boldsymbol{\Psi}_l^{-1}(\hat{\boldsymbol{\theta}}) & \mathbf{0} \\ \boldsymbol{C}_l & \boldsymbol{\Psi}_l^{-1}(\hat{\boldsymbol{\theta}}) \end{bmatrix} \tag{39}$$

where $\boldsymbol{C}_l$ is equal to $\boldsymbol{C}_r$ except for a change of sign for the $\frac{1}{2}\hat{\boldsymbol{\rho}}$ term. It is seen from the Taylor series expansions

$$\frac{1-\alpha(\theta)}{\theta^2} = \frac{1}{12} + \frac{\theta^2}{720} + \frac{\theta^4}{30\,240} + \dots \tag{40}$$

$$\frac{\alpha(\theta)+\beta(\theta)-2}{\theta^4} = -\frac{1}{720} - \frac{\theta^2}{15\,120} - \frac{\theta^4}{403\,200} + \dots \tag{41}$$

that the coefficients in (38) are well-behaved for all θ.

IV. The Lie Algebraic UKF on SE(3)

A. System Dynamics

The state is given by $\boldsymbol{T} \in SE(3)$, and the system dynamics is the kinematic differential equation

$$\dot{\boldsymbol{T}} = \boldsymbol{T}[\boldsymbol{V} + \boldsymbol{w}]^{\wedge}_{SE(3)} \tag{42}$$

where $\boldsymbol{V}$ is the vector form of the right velocity, and $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ is a noise vector. The discrete-time model is

$$\boldsymbol{T}_{k+1} = \boldsymbol{T}_k \exp([\boldsymbol{\xi}_{k+1}]^{\wedge}_{SE(3)}) \tag{43}$$

where the global state is given by $\boldsymbol{T}_k$, and the increment from one sample to the next is described by the logarithm $\boldsymbol{\xi}_{k+1}$. This is similar to the multiplicative EKF [15] where a quaternion gives the global state, and a 3-dimensional vector is used for the increment. The corresponding discrete-time equation for the logarithm is

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + h\boldsymbol{\Phi}_r^{-1}\left(\mathbf{ad}(\boldsymbol{\xi}_k)\right)\left(\boldsymbol{V}_k + \boldsymbol{w}_k\right) \tag{44}$$

where $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k)$. It is assumed that we can measure the full pose, and that the measurements are given by

$$\boldsymbol{Z} = \boldsymbol{T}\exp([\boldsymbol{\nu}]^{\wedge}_{SE(3)}) \ \in SE(3) \tag{45}$$

where $\boldsymbol{\nu}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{N}_k)$ is the measurement noise vector. In the proposed UKF the kinematic differential equation (44) is used to describe the system dynamics, whereas previous work used the kinematic differential equation (43) on G.

B. Filter Dynamics

The sigma points of the time update are given by $\boldsymbol{\xi}^a_{k|k,i} = [\boldsymbol{\xi}^{x\ \mathrm{T}}_{k|k,i}, \boldsymbol{\xi}^{w\ \mathrm{T}}_{k|k,i}]^{\mathrm{T}}$, $i = 0, \ldots, 2m$ where $m = 12$. The vector $\boldsymbol{\xi}^x_i \in \mathbb{R}^6$ is related to the state variables, and $\boldsymbol{\xi}^w_i \in \mathbb{R}^6$ correspond to the process noise. The sigma points are propagated by using the discretized dynamics in (44), which gives

$$\boldsymbol{\xi}^x_{k+1|k,i} = \boldsymbol{\xi}^x_{k|k,i} + h\boldsymbol{\Phi}_r^{-1}(\mathbf{ad}(\boldsymbol{\xi}^x_{k|k,i}))(\boldsymbol{V}_{m,k} + \boldsymbol{\xi}^w_{k|k,i}) \tag{46}$$

where $\boldsymbol{V}_{m,k}$ are the measured velocities. The predicted mean of the logarithm is computed as the weighted sum

$$\bar{\boldsymbol{\xi}}_{k+1|k} = \sum_{i=0}^{2q} w_i^{\mu}\boldsymbol{\xi}^x_{k+1|k,i} \quad \in \mathbb{R}^6 \tag{47}$$

where the weights w_i^{μ} are given in Step 2 in Algorithm 1. This is used to calculate the predicted mean of the global state as

$$\boldsymbol{T}_{k+1|k} = \boldsymbol{T}_{k|k}\exp([\bar{\boldsymbol{\xi}}_{k+1|k}]^{\wedge}_{SE(3)}) \tag{48}$$

and the deviation $\boldsymbol{e}_{k+1|k,i} = \boldsymbol{\xi}^x_{k+1|k,i} - \bar{\boldsymbol{\xi}}_{k+1|k}$. Then, using the result (11), the covariance is computed from

$$\boldsymbol{P}_{k+1|k} = \sum_{i=0}^{2m} w_i^P(\boldsymbol{\epsilon}_{k+1|k,i})(\boldsymbol{\epsilon}_{k+1|k,i})^{\mathrm{T}} \tag{49}$$

where $\boldsymbol{\epsilon}_{k+1|k,i} = \Phi_r(\mathbf{ad}(\bar{\boldsymbol{\xi}}_{k+1|k}))\boldsymbol{e}_{k+1|k,i}$ and the weights w_i^P are given in Step 2 in Algorithm 1. Figure 1 illustrates how an error vector on the logarithm, $\boldsymbol{e}$, maps to an error vector, $\boldsymbol{\epsilon}$, on the tangent plane shifted by the exponential map of the mean vector, $\bar{\boldsymbol{\xi}}$.

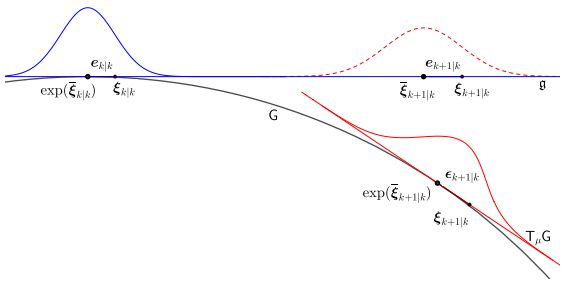

Fig. 1: The relation between a predicted error vector $\boldsymbol{e}_{k+1|k}$ propagated on the tangent space at the identity, since $\bar{\boldsymbol{\xi}}_{k|k} = \boldsymbol{0}$, and the corresponding error vector $\boldsymbol{\epsilon}_{k+1|k}$ on the tangent space at the predicted state, which is given through the Jacobian.

The new contribution is that the sigma points are computed from (46) with the kinematic differential equation of the logarithm. Then the mean of the time update is computed from (47) as the average of vectors, while the covariance is calculated in terms of vectors on the tangent plane and is transformed to the next tangent plane by a matrix operation. This is computationally more efficient than averaging on G, which is typically done in previous work [23], [22], [32], [36], or computing the time propagation on G and then averaging the logarithm and using parallel transport of the covariance as in [31]. In [22] it was stated that the mean and the covariance can be computed in terms of the logarithm, and that the covariance could be transformed with parallel transport, however, details on the time propagation of the sigma points and the parallel were not included. In [10] the mean was not calculated from the sigma points, instead, based on [3], velocity measurements were used to calculate the mean. Moreover, we would like to point out that the proposed formulation gives a UKF for Lie groups that is more similar to the original formulation on $\mathbb{R}^n$, which could facilitate ease of understanding and implementation.

C. Measurement Update

The measurement update is based on the formulation of [10] and is computed according to Algorithm 2. The sigma points of the measurement update are given by $\boldsymbol{\xi}^a_{k+1|k,i} = [\boldsymbol{\xi}^{x\ \mathrm{T}}_{k+1|k,i}, \boldsymbol{\xi}^{\nu\ \mathrm{T}}_{k+1|k,i}]^{\mathrm{T}}$. The logarithm $[\boldsymbol{\zeta}_{k+1,i}]^{\wedge}$ corresponding to sigma point i of the measurement is defined by

$$\exp\left([\boldsymbol{\zeta}_{k+1,i}]^{\wedge}\right) = \exp\left([\boldsymbol{\xi}^x_{k+1|k,i}]^{\wedge}\right)\exp\left([\boldsymbol{\xi}^{\nu}_{k+1,i}]^{\wedge}\right) \tag{50}$$

The weighted mean is found from $\bar{\boldsymbol{\zeta}}_{k+1} = \sum_{i=0}^{2m} w_i^{\mu}\boldsymbol{\zeta}_{k+1,i}$ where w_i^{μ} is found in Step 2 in Algorithm 2. The covariance can then be computed from $\Delta\boldsymbol{\zeta}_{k+1,i} = \boldsymbol{\zeta}_{k+1,i} - \bar{\boldsymbol{\zeta}}_{k+1}$. The remaining steps are given in Algorithm 2. It is noted that the transformation of the covariance in Step 15 of Algorithm 2 was not performed in [10], but appeared in [8] for an EKF on Lie groups.

The calculations of the measurement update are in terms of vectors in the tangent plane, which simplifies implementation, and can potentially reduce computational costs. This method

Algorithm 1 UKF-LieAlg Time Update on $SE(3)$

Input: $\boldsymbol{T}_{k|k}, \boldsymbol{P}_{k|k}, \boldsymbol{Q}_k, \boldsymbol{V}_k, h, \alpha, \beta$

$\lambda = (\alpha^2 - 1)m$ ▷ $m = 12$

$w_0^\mu = \frac{\lambda}{\lambda+m},\ w_{j>0}^\mu = \frac{1}{2(\lambda+m)}$

$w_0^P = \frac{\lambda}{\lambda+m} + 1 - \alpha^2 + \beta,\ w_{j>0}^P = \frac{1}{2(\lambda+m)}$ ▷ $\beta = 2$

$\boldsymbol{P}_{k|k}^a = \text{diag}(\boldsymbol{P}_{k|k}, \boldsymbol{Q}_k)$

$\boldsymbol{\sigma}_{k|k} = \text{Chol}((m+\lambda)\boldsymbol{P}_{k|k}^a)$

$\boldsymbol{\xi}_{k|k,0}^a = \boldsymbol{0} \in \mathbb{R}^m$

$\boldsymbol{\xi}_{k|k,i}^a = \text{col}_i(\boldsymbol{\sigma}_{k|k})$ ▷ $i = 1 \dots m$

$\boldsymbol{\xi}_{k|k,i+m}^a = -\text{col}_i(\boldsymbol{\sigma}_{k|k})$

$\boldsymbol{\xi}_{k|k,i}^a = [\boldsymbol{\xi}_{k|k,i}^{x\,\mathrm{T}},\ \boldsymbol{\xi}_{k|k,i}^{w\,\mathrm{T}}]^\mathrm{T}$ ▷ $\boldsymbol{\xi}_i^x, \boldsymbol{\xi}_i^w \in \mathbb{R}^6$

$\boldsymbol{\xi}_{k+1|k,i}^x = \boldsymbol{\xi}_{k|k,i}^x + h\boldsymbol{\Phi}_r^{-1}(\mathbf{ad}(\boldsymbol{\xi}_{k|k,i}^x))(\boldsymbol{V}_k + \boldsymbol{\xi}_{k|k,i}^w)$

$\bar{\boldsymbol{\xi}}_{k+1|k} = \sum_{i=0}^{2m} w_i^\mu \boldsymbol{\xi}_{k+1|k,i}^x$

$\boldsymbol{e}_{k+1|k,i} = \boldsymbol{\xi}_{k+1|k,i}^x - \bar{\boldsymbol{\xi}}_{k+1|k}$

$\boldsymbol{P}_{k+1|k} = \boldsymbol{J}_1 \left[\sum_{i=0}^{2m} w_i^P (\boldsymbol{e}_{k+1|k,i})(\boldsymbol{e}_{k+1|k,i})^\mathrm{T}\right] \boldsymbol{J}_1^\mathrm{T}$ ▷ $\boldsymbol{J}_1 = \boldsymbol{\Phi}_r(\mathbf{ad}(\bar{\boldsymbol{\xi}}_{k+1|k}))$

$\boldsymbol{T}_{k+1|k} = \boldsymbol{T}_{k|k} \exp([\bar{\boldsymbol{\xi}}_{k+1|k}]^\wedge)$

Output: $\boldsymbol{T}_{k+1|k}, \boldsymbol{P}_{k+1|k}$

was used in [10], while [22], [23], [32], [36] used Lie group elements which were transformed to the tangent plane.

Algorithm 2 UKF-LieAlg Measurement Update on $SE(3)$

Input: $\boldsymbol{T}_{k+1|k}, \boldsymbol{P}_{k+1|k}, \boldsymbol{Z}_{k+1}, \boldsymbol{N}_k \in \mathbb{R}^{6\times6}$

$\lambda = (\alpha^2 - 1)r$ ▷ $r = 12$

$w_0^\mu = \frac{\lambda}{\lambda+r},\ w_{j>0}^\mu = \frac{1}{2(\lambda+r)}$

$w_0^P = \frac{\lambda}{\lambda+r} + 1 - \alpha^2 + \beta,\ w_{j>0}^P = \frac{1}{2(\lambda+r)}$ ▷ $\beta = 2$

$\boldsymbol{P}_{k+1|k}^a = \text{diag}(\boldsymbol{P}_{k+1|k}, \boldsymbol{N}_k)$

$\boldsymbol{\sigma}_{k+1|k} = \text{Chol}((r+\lambda)\boldsymbol{P}_{k|k}^a)$

$\boldsymbol{\xi}_{k+1|k,0}^a = \boldsymbol{0} \in \mathbb{R}^r$

$\boldsymbol{\xi}_{k+1|k,i}^a = \text{col}_i(\boldsymbol{\sigma}_{k+1|k})$ ▷ $i = 1 \dots r$

$\boldsymbol{\xi}_{k+1|k,i+r}^a = -\text{col}_i(\boldsymbol{\sigma}_{k+1|k})$

$\boldsymbol{\xi}_{k+1|k,i}^a = \left[\boldsymbol{\xi}_{k+1|k,i}^{x\,\mathrm{T}},\ \boldsymbol{\xi}_{k+1|k,i}^{\nu\,\mathrm{T}}\right]^\mathrm{T}$ ▷ $\boldsymbol{\xi}^\nu \in \mathbb{R}^6$

$\boldsymbol{\zeta}_{k+1,i} = \left[\log\left(\exp([\boldsymbol{\xi}_{k+1|k,i}^x]^\wedge)\exp([\boldsymbol{\xi}_{k+1,i}^\nu]^\wedge)\right)\right]^\vee$

$\bar{\boldsymbol{\zeta}}_{k+1} = \sum_{i=0}^{2m} w_i^\mu \boldsymbol{\zeta}_{k+1,i}$

$\Delta\boldsymbol{\zeta}_{k+1,i} = \boldsymbol{\zeta}_{k+1,i} - \bar{\boldsymbol{\zeta}}_{k+1}$

$\boldsymbol{P}_{xz} = \sum_{i=0}^{2m} w_i^\mu (\boldsymbol{\xi}_{k+1|k,i}^x)(\Delta\boldsymbol{\zeta}_{k+1,i})^\mathrm{T}$

$\boldsymbol{P}_{zz} = \sum_{i=0}^{2m} w_i^\mu (\Delta\boldsymbol{\zeta}_{k+1,i})(\Delta\boldsymbol{\zeta}_{k+1,i})^\mathrm{T}$

$\boldsymbol{P}_{k+1|k+1}^- = \boldsymbol{P}_{k+1|k} - \boldsymbol{K}\boldsymbol{P}_{zz}\boldsymbol{K}^\mathrm{T}$ ▷ $\boldsymbol{K} = \boldsymbol{P}_{xz}\boldsymbol{P}_{zz}^{-1}$

$\boldsymbol{\eta}_{k+1} = \left[\log(\boldsymbol{T}_{k+1|k}^{-1}\boldsymbol{Z}_{k+1})\right]^\vee$

$\boldsymbol{m}_{k+1} = \boldsymbol{K}\boldsymbol{\eta}_{k+1}$

$\boldsymbol{P}_{k+1|k+1} = \boldsymbol{J}_2\boldsymbol{P}_{k+1|k+1}^-\boldsymbol{J}_2^\mathrm{T}$ ▷ $\boldsymbol{J}_2 = \boldsymbol{\Phi}_r(\mathbf{ad}(\boldsymbol{m}_{k+1}))$

$\boldsymbol{T}_{k+1|k+1} = \boldsymbol{T}_{k+1|k} \exp([\boldsymbol{m}_{k+1}]^\wedge)$

Output: $\boldsymbol{T}_{k+1|k+1}, \boldsymbol{P}_{k+1|k+1}$

V. Simulations

In this section, we present a comparison of 3 UKF filters: The UKF-LG of [10], our proposed Lie algebraic UKF (UKF-LA) described in IV, and our method with optimization on the manifold in the prediction step (UKF-LA-Opt) according to (18). The parameter $\alpha = 10^{-3}$ was used in all cases. The velocity measurements were body-fixed and obtained at a rate of 100 Hz, while the pose measurements were given in the inertial frame and obtained with a sample rate of 1 Hz. Furthermore, the measurements were elements of $SE(3)$, and the measurement noise was multiplicative and assumed to be given as in (45). The angular and linear velocities in the time interval $t \in [0, T]$ were given by $\boldsymbol{\omega}^b(t) = [0, 0, 2t/(0.001t^2 + 8t + 1)]^\mathrm{T}$ rad/s and $\boldsymbol{v}^b(t) = [t/(1+2t), 0, 0]^\mathrm{T}$ m/s, and were used to describe the right velocity as given in (30). The analytic expressions allowed for exact solutions, and we could therefore use an exact true trajectory for comparisons. All simulations were initiated with the covariance $\boldsymbol{P}_{0|0} = 0.01\boldsymbol{I}_{6\times6}$. The process noise matrix was given as $\boldsymbol{Q} = \text{diag}(\boldsymbol{Q}_\omega, \boldsymbol{Q}_v)$ where $\boldsymbol{Q}_\omega = \sigma_\omega^2\boldsymbol{I}_{3\times3}$ and $\boldsymbol{Q}_v = \sigma_v^2\boldsymbol{I}_{3\times3}$, which describes the uncertainty of the velocity measurements. The noise parameters were set to $\sigma_\omega = 0.1$ rad/s, and $\sigma_v = 0.05$ m/s. The covariance matrix describing the measurement noise matrix was given as $\boldsymbol{N} = \text{diag}(\boldsymbol{N}_R, \boldsymbol{N}_p)$ where $\boldsymbol{N}_R = \sigma_R^2\boldsymbol{I}_{3\times3}$ and $\boldsymbol{N}_p = \sigma_p^2\boldsymbol{I}_{3\times3}$. The measurement noise parameters were given as $\sigma_R = \frac{\pi}{180}$ rad and $\sigma_p = 0.01$ m. The angular error at time t_k was computed as $\theta_{e,k} = \|[\log(\boldsymbol{R}_{k|k}^\mathrm{T}\boldsymbol{R}_k)]_{SO(3)}^\vee\|$ where $\boldsymbol{R}_{k|k}$ is the estimated rotation matrix between the body-fixed frame and the inertial frame, and $\boldsymbol{R}_k$ was the true attitude of the system. The positional error was found as $r_{e,k} = \|\boldsymbol{r}_{k|k} - \boldsymbol{r}_k\|$ where $\boldsymbol{r}_{k|k}$ was the estimated position and $\boldsymbol{r}_k$ was the true position, both given in the inertial frame.

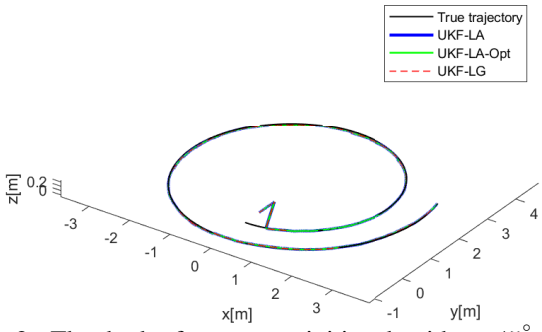

Fig. 2: The body frame was initiated with a 45° angular error as well as 0.5 meters off the initial position. Before any measurement updates had been obtained the predicted motion was incorrect and the body frame was heading away from the xy plane which the true trajectory reside on, but was able to converge towards the true trajectory as the pose was measured.

A. Case Studies

In the first case the trajectory was estimated over $T = 40$ seconds. The trajectory was in the xy plane. The estimator had an initial state given by a rotation of 45° about the y axis, and by a positional offset of 0.5 m along the y axis. Due to the initial angular offset, estimated state left the xy plane for the first 100 samples (Figure 2), until the first pose measurement made the estimator errors converge to values close to zero, as seen in Figure 3.

In the second case, the three estimators had initial states given by identity matrices, such that $\boldsymbol{T}_{0|0} = \boldsymbol{I}_{4\times4}$ before

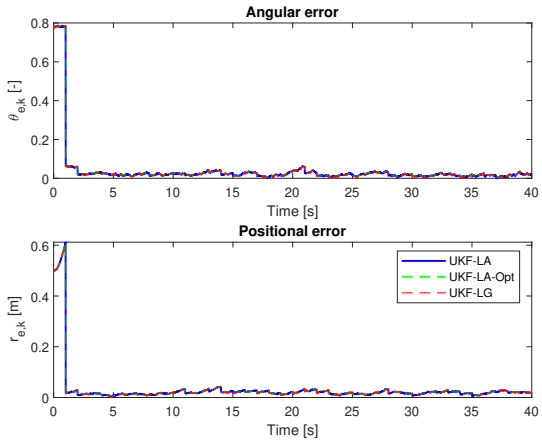

Fig. 3: The estimation error due to a poorly chosen initial condition was reduced when the pose of the body had been measured.

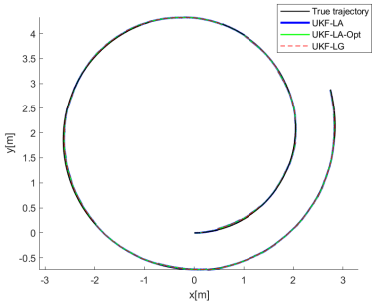

Fig. 4: Spiral trajectory initiated with zero offsets.

estimating the trajectory. It is seen in Figure 4 that the all the three estimators tracked the trajectory with high accuracy. The estimates provided by the three filters were close to indistinguishable when they are used with the same set of measurements. This is seen in Figures 3 and 5.

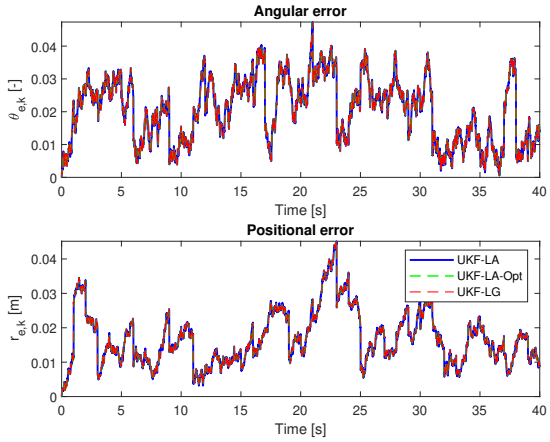

Fig. 5: Angular and positional error compared with the ground truth when the system after the system had been initialized with zero offsets.

B. Computational efficiency

In order to evaluate the computational effort of the UKF filters, 100 different sets of full pose measurements of 1000 samples were generated, where the time update was performed at a rate of 100 Hz, and the measurement update was performed at 1 Hz. The computational time over each set of measurements was evaluated for each estimator, and the mean computational time was computed. The average computational time spent for each estimator is provided in Table I together with the difference given in terms of percentages. It is seen that the proposed UKF-LA gave computational time which was 37.3 % of the computational time of UKF-LG, which was set to 100 %. When the mean was obtained through optimization on the manifold, then UKF-LA was slower than UKF-LG. It is noted that closed form solutions of the exponential and logarithmic maps on $SE(3)$, as described in Equation (26) and (27), were used for efficiency in computation. If library functions in MATLAB were used for computation of the exponentials and the logarithms maps, then UKF-LA could perform up to 10 times faster than UKF-LG. This becomes evident when studying the prediction step presented in [10], as the exponential map must be computed twice, the logarithmic map once, and two matrix multiplications are required for each sigma point in each prediction step. In contrast, the exponential map is computed once per prediction step in UKF-LA, and no logarithms must be called unless the estimated mean is obtained through optimization.

UKF-LA	UKF-LG	UKF-LA-Opt
0.6117 s	1.6385 s	1.9220 s
37.3 %	100 %	117.3 %

TABLE I: The UKF-LA required less computational effort compared to UKF-LG and UKF-LA-Opt.

VI. Conclusion and Future Works

A UKF for matrix Lie groups has been proposed where the time propagation is formulated in terms of the kinematic differential equation of the logarithm. The proposed method is to a large extent formulated in terms of vector operations on the Lie algebra, and the formulation is closer to the original UKF on $\mathbb{R}^n$ than previous works on Lie groups. This leads to efficient formulations and potentially to reduced computational costs, in particular in the time update. The paper includes details on how to implement the proposed UKF for $SE(3)$. The method was compared to the UKF-LG of [10] in simulations on $SE(3)$, where it was found that the difference in the estimator errors was not significant, while the computational cost of the proposed method was 37 % of the UKF-LG. The proposed method with averaging of the sigma points on the Lie algebra was compared to a method with averaging of the sigma points on the manifold. Again, there was no significant difference in the estimation error, and the computation time of the proposed method was 32 % of the method with averaging on the manifold. Future work may include equations of motion and different sensor systems including IMUs and bias modeling.

Acknowledgment

The research presented in this paper was funded by the Norwegian Research Council under Project Number 237896, SFI Offshore Mechatronics.

REFERENCES

[1] G. Baldwin, R. E. Mahony, J. Trumpf, T. Hamel, and T. Cheviron. Complementary filter design on the special Euclidean group SE(3). *2007 European Control Conference (ECC)*, pages 3763–3770, 2007.

[2] F. Barbaresco. Lie group statistics and Lie group machine learning based on Souriau Lie groups thermodynamics & Koszul-Souriau-Fisher metric: New entropy definition as generalized Casimir invariant function in coadjoint representation. *Entropy*, 22(6), 2020.

[3] T. D. Barfoot and P. T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014.

[4] A. Barrau and S. Bonnabel. The invariant extended Kalman filter as astable observer. *IEEE Transactions on Automatic Control*, 62(4):1797–1812, 2017.

[5] S. Bonnabel, P. Martin, and P. Rouchon. Symmetry-preserving observers. *IEEE Transactions on Automatic Control*, 53(11):2514–2526, 2008.

[6] S. Bonnabel, P. Martin, and P. Rouchon. Non-linear symmetry-preserving observers on Lie groups. *IEEE Transactions on Automatic Control*, 54(7):1709–1713, 2009.

[7] G. Bourmaud, R. Mégret, M. Arnaudon, and A. Giremus. Continuous-discrete extended Kalman filter on matrix Lie groups using concentrated Gaussian distributions. *Journal of Mathematical Imaging and Vision*, 51(1):209–228, 2015.

[8] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu. Discrete extended Kalman filter on Lie groups. *21st European Signal Processing Conf. (EUSIPCO)*, 2013.

[9] M. Brossard, A. Barrau, and S. Bonnabel. Exploiting symmetries to design EKFs with consistency properties for navigation and SLAM. *IEEE Sensors Journal*, 19(4):1572–1579, Feb 2019.

[10] M. Brossard, S. Bonnabel, and J.-P. Condomines. Unscented Kalman filtering on Lie groups. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2485–2491, 2017.

[11] F. Bullo and R. M. Murray. Proportional Derivative (PD) Control on the Euclidean Group. CDS Technical Report 95-010, California Institute of Technology, 1995.

[12] J. Ćesić, I. Marković, M. Bukal, and I. Petrović. Extended information filter on matrix Lie groups. *Automatica*, 82:226–234, 2017.

[13] J. Ćesić, I. Marković, and I. Petrović. Mixture reduction on matrix Lie groups. *IEEE Signal Processing Letters*, 24(11):1719–1723, 2017.

[14] G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups*, volume 2. Birkhäuser, 2012.

[15] J. Crassidis and F. L. Markley. Unscented filtering for spacecraft attitude estimation. *Journal of Guidance Control and Dynamics*, 26(4):536–542, 2003.

[16] J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.

[17] Y. Deng, Z. Wang, and L. Liu. Unscented Kalman filter for spacecraft pose estimation using twistors. *Journal of Guidance, Control, and Dynamics*, 39(8):1844–1856, 2016.

[18] J. Duistermaat and J. Kolk. *Lie Groups*. Springer, New York, 2000.

[19] O. Egeland and J.-M. Godhavn. Passivity-based adaptive attitude control of a rigid spacecraft. *IEEE Transactions on Automatic Control*, 39(4):842–845, 1994.

[20] N. Filipe, M. Kontitsis, and P. Tsiotras. Extended Kalman filter for spacecraft pose estimation using dual quaternions. *Journal of Guidance, Control, and Dynamics*, 38(9):1–17, 2015.

[21] B. C. Hall. *Lie Groups, Lie Algebras, and Representations. An Elementary Introduction*. Graduate Texts in Mathematics. Springer, Berlin, Heidelberg, New York, 2003.

[22] S. Hauberg, F. Lauze, and K. Pedersen. Unscented Kalman filtering on Riemannian manifolds. *Journal of Mathematical Imaging and Vision*, 46:103–120, 2013.

[23] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 14(1):57–77, 2013.

[24] M.-D. Hua, M. Zamani, J. Trumpf, R. E. Mahony, and T. Hamel. Observer design on the special Euclidean group SE(3). *IEEE Conference*

[26] S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Defense, Security, and Sensing*, 1997.

on Decision and Control and European Control Conference, pages 8169–8175, 2011.

[25] A. Iserles, H. Munthe-Kaas, S. Nørsett, and A. Zanna. Lie-group methods. *Acta Numerica*, 2005.

[27] S. Klarsfeld and J. A. Oteo. The Baker-Campbell-Hausdorff formula and the convergence of the Magnus expansion. *J. Phys. A: Math. Gen.*, 22:4565–4572, 1989.

[28] C. Lageman, J. Trumpf, and R. Mahony. Gradient-like observers for invariant dynamics on a Lie group. *IEEE Transaction on Automatic Control*, 55(2):367–377, 2010.

[29] T. Lee, M. Leok, and N. McClamroch. Global symplectic uncertainty propagation on $SO(3)$. *Proc. IEEE Conf. on Decision and Control*, pages 61–66, 2008.

[30] E. J. Lefferts, F. L. Markley, and M. D. Shuster. Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429, 1982.

[31] G. Loianno, M. Watterson, and V. Kumar. Visual inertial odometry for quadrotors on SE(3). In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 1544–1551. IEEE, May 2016.

[32] G. Magalhães, Y. Cáceres, J. B. do Val, and R. S. Mendes. UKF on Lie groups for radar tracking using polar and Doppler measurements. In *XXII Congresso Brasileiro de Automática*, 09 2018.

[33] R. Mahony, T. Hamel, P. Morin, and E. Malis. Nonlinear complementary filters on the special linear group. *International Journal of Control*, 85(10):1557–1573, 2012.

[34] R. Mahony, T. Hamel, and J.-M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, 2008.

[35] H. M. Menegaz, J. Y. Ishihara, G. de Araújo Borges, and A. N. Vargas. A systematization of the unscented Kalman filter theory. *IEEE Transactions on Automatic Control*, 60(10):2583–2598, 2015.

[36] H. M. T. Menegaz, J. Y. Ishihara, and H. T. M. Kussaba. Unscented Kalman filters for Riemannian state-space systems. *IEEE Transactions on Automatic Control*, 64(4):1487–1502, 2019.

[37] F. C. Park. Distance metrics on the rigid-body motions with applications to mechanism design. *J. Mechanical Design*, 117(1):48–54, 1995.

[38] M. E. Pittelkau. Rotation vector in attitude estimation. *Journal of Guidance, Control, and Dynamics*, 26(6):855–860, 2003.

[39] H. Rehbinder and B. K. Ghosh. Pose estimation using line-based dynamic vision and inertial sensors. *IEEE Transactions on Automatic Control*, 48(2):186–199, 2003.

[40] S. Salcudean. A globally convergent angular velocity observer for rigid body motion. *IEEE Transactions on Automatic Control*, 36(12):1493–1497, 1991.

[41] B. J. Sipos. Application of the manifold-constrained unscented Kalman filter. In *Proceedings 2008 IEEE/ION Position, Location and Navigation Symposium*, pages 30–43. IEEE, 2008.

[42] A. M. Sjøberg and O. Egeland. An EKF for Lie groups with application to crane load dynamics. *Modeling, Identification and Control*, 40(2):109–124, 2019.

[43] R. A. Srivatsan, M. Xu, N. Zevallos, and H. Choset. Probabilistic pose estimation using a Bingham distribution-based linear filter. *Int. J. Robotics Res.*, 37(13-14), 2018.

[44] A. Sveier, A. M. Sjøberg, and O. Egeland. Applied Runge–Kutta–Munthe-Kaas integration for the quaternion kinematics. *Journal of Guidance Control and Dynamics*, 42(12):2747–2754, 2019.

[45] J. Thienel and R. M. Sanner. A coupled nonlinear spacecraft attitude controller and observer with an unknown constant gyro bias and gyro noise. *IEEE Transaction on Automatic Control*, 48(11):2011–2015, 2003.

[46] B. Vik and T. I. Fossen. A nonlinear observer for GPS and INS integration. In *Proceedings of the 40th IEEE Conference on Decision and Control*, page 2956–2961, Dec 2001.

[47] Y. Wang and G. S. Chirikjian. Error propagation on the Euclidean group with applications to manipulator kinematics. *IEEE Transactions on Robotics*, 22(4):591–602, 2006.

[48] J. T. Wen and K. Kreutz-Delgado. The attitude control problem. *IEEE Transactions on Automatic Control*, 36(10):1148–1162, 1991.