An Introduction to the Invariant Extended Kalman Filter

EASTON R. POTOKAR, RANDAL W. BEARD, AND JOSHUA G. MANGELSON

Kalman filters (KFs) have become ubiquitous in modern-day technology, with applications in medicine, robotics, and many other areas, due to their straightforward implementation and impressive performance. The KF is the optimal solution in the mean-square-error sense for tracking a linear stochastic state [1] and is well-known for its asymptotic convergence guarantee [2]. The extension of the KF to nonlinear systems is the extended KF (EKF), which has been the de facto standard for many years, even though it lacks the theoretical guarantees of the standard KF. Other KF flavors have been introduced, including the unscented KF [3] and the error state KF (ESKF) [4], all of which have their limitations and tradeoffs. The most glaring disadvantage for all these KF variants is the state-dependent linearization that can cause poor performance and weak convergence.

The recent development of the invariant EKF (InEKF) solves this problem for a certain class of systems by defining an error invariant to a class of symmetries [5], [6]. The error is defined by composition on a Lie group, which in our examples will be matrix multiplication, and results in the estimation error satisfying a log-linear differential equation, state-independent error system dynamics, and a local asymptotic convergence guarantee, which the EKF lacks. More recently, the class of systems that fit the InEKF paradigm has been more clearly defined and is known as group affine systems [7], [8]. The InEKF has led to numerous successful results and applications in simultaneous localization and mapping (SLAM) [9], [10], [11], visual–inertial navigation [12], [13], [14], guided aerial navigation [9], [15], [16], [17], 3D bipedal robots [18], [19], and underwater navigation [20]. Additionally, there are many other theoretical developments of the InEKF with regard to observability [21], reachability [22], [23], inertial measurement unit (IMU) preintegration [24], and usage in a multistate constrained KF framework [25].

Unfortunately, the current literature describing the InEKF requires a deep theoretical understanding of Lie groups and is not accessible enough to a general audience to allow more widespread adoption. Our purpose in this article is to fill this gap in the literature by providing an accessible introduction to the necessary theoretical background, focusing on implementation choices and details, and by providing several simple examples with full derivations (see "Summary" for an overview). Additionally, there are many scenarios when the tracked state is not completely group affine; in these scenarios, an "imperfect" InEKF can be still be used, which, while it lacks the convergence guarantees, still exhibits improved performance over other EKF formulations. We illustrate with an example of an imperfect InEKF, again with detailed derivations. We note that that our treatment of the InEKF is not comprehensive of all state-of-the-art developments but, rather, tutorial in nature. We focus on introducing the necessary theory to utilize the InEKF as well as some practical choices for implementation purposes.

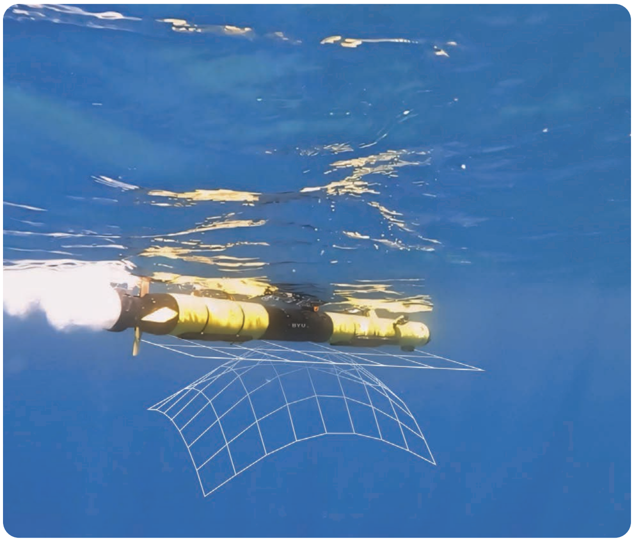
IMAGE COURTESY OF THE BYU FROSTLAB

Since the error is based upon noncommutative matrix multiplication, the InEKF has both right and left variants, and the difference between these variants has limited discussion in the current literature. While both variants perform similarly, there are nuances that can cause one to perform better than the other, depending on the application. In much of the existing literature, the right or left formulation of the InEKF is selected with a limited discussion based on the specific application [8], [26], [27], [28]. One contribution of this article is to shed additional light on the differences between the two variants, including a detailed

Digital Object Identifier 10.1109/MCS.2024.3466488
Date of current version: 14 November 2024

1066-033X/24©2024IEEE

description of their nuances and an explanation of which variant will perform better for a given application.

Specific contributions of this article are as follows:

1) We provide step-by-step tutorials of deriving both the right and left InEKFs together with a detailed description of when one should be selected over the other. We provide a similar tutorial for the imperfect InEKF.
2) We provide several simple examples showing the application of the InEKF and the imperfect InEKF to specific problems.
3) We provide an all-inclusive open source C++ and Python library for easy implementation of both the right and left InEKFs, with Lie group structures and flexible measurement models already built in.

Summary

Successfully and accurately estimating the 3D position of a vehicle is a necessary step for any form of robotic autonomy. In a linear setting, there are straightforward optimal (in the mean-square-error sense) solutions, such as the KF, that provide out-of-the-box methods for estimation. Unfortunately, most systems in the real world are highly nonlinear, and while there are many formulations of the KF that empirically work well with nonlinear systems, the KF convergence guarantees do not apply to general nonlinear systems.

For systems tracking states that meet a minimal form of symmetry, such as 3D rotations, the InEKF provides a state-independent linearization that provides convergence guarantees similar to the KF. This article provides an in-depth introduction to and tutorial on the InEKF by covering the necessary mathematical background, providing derivations of the InEKF's various components, and walking step by step through various examples of its usage. Additionally, an open source library is provided for straightforward implementation under a variety of process and measurement models.

This article proceeds as follows. First, we give the necessary mathematical background on matrix Lie groups, followed by a review of the EKF and ESKF and then a derivation of the InEKF. The nuances between the right and left InEKF error, prediction step, and update step are then laid out and explained. Step-by-step examples with derivations and decision factors are then given for a simple odometry model as well as an imperfect InEKF for an inertial model. The article then concludes and discusses possible research directions.

MATRIX LIE GROUPS

In this section, we provide a brief introduction to matrix Lie groups as well as various mathematical fundamentals to help in understanding the InEKF. See "Groups and Smooth Manifolds" for additional details about groups and manifolds.

A Lie group is a group and a smooth manifold with the property that element inversion and the group operation are differentiable [29]. A matrix Lie group is simply a Lie

Groups and Smooth Manifolds

GROUPS

A group is defined as any set $\mathcal{G}$ along with an operation $\cdot : \mathcal{G}\times\mathcal{G}\rightarrow\mathcal{G}$ that satisfies the following properties $\forall a,b,c\in\mathcal{G}$ [S1]:

- *Closure*: $a\cdot b\in\mathcal{G}$.
- *Associativity*: $(a\cdot b)\cdot c = a\cdot(b\cdot c)$.
- *Identity*: There exists an $I\in\mathcal{G}$ such that $I\cdot a = a\cdot I = a$.
- *Inverse*: There exists an $a^{-1}\in\mathcal{G}$ such that $a\cdot a^{-1} = a^{-1}\cdot a = I$.

Groups are generally an abstraction of symmetry. That is, each element in $\mathcal{G}$ can be thought of as a transformation that leaves a certain object unchanged. For example, a group of rotations of a square could be given by the set $G=\{0,\pi/2,\pi,3\pi/2\}$. It is straightforward to verify that each of the above properties holds on this set with the addition-modulo-2π operation.

SMOOTH MANIFOLDS $\mathbb{R}^n$

A smooth manifold of dimension n is a geometric structure that is locally close to $\mathbb{R}^n$, enabling the use of calculus on a local map of the manifold [29]. Examples include the unit sphere, a torus, or a smooth surface in $\mathbb{R}^3$. Manifolds are often visualized by a curved surface, where each point $X\in\mathcal{M}$ is equipped with a tangent space denoted $T_X\mathcal{M}$, as shown in Figure 1. Each tangent space is a vector space of dimension n and can be isomorphically mapped to $\mathbb{R}^n$ [30] using the "hat" operator, which is denoted as

$$^\wedge : \mathbb{R}^n \rightarrow T_X\mathcal{M}. \tag{S1}$$

Each element of the tangent space is called a vector and often denoted by $\xi\in T_X\mathcal{M}$. Furthermore, a map exists from $T_X\mathcal{M}$ to a neighborhood of $X\in\mathcal{M}$, called the exponential map:

$$\exp_X : T_X\mathcal{M}\rightarrow\mathcal{M}. \tag{S2}$$

In some cases, exp is the well-known exponential given by the Taylor series $e^X=\Sigma_{n=0}^{\infty}X^n/n!$, but generally, it is not [29].

REFERENCE

[S1] T. Rowland and E. W. Weisstein, "Group," Wolfram. 2022. Accessed: Oct. 1, 2024. [Online]. Available: https://mathworld.wolfram.com/Group.html

group whose elements are matrices, and the group operation is matrix multiplication.

The Lie algebra $\mathfrak{g}$ of a Lie group $\mathcal{G}$ is the tangent space to the smooth manifold $\mathcal{G}$ at the identity; thus, $\mathfrak{g} = T_I\mathcal{G}$ [30], as shown in Figure 1. In the case of a matrix Lie group, the identity is the identity matrix. The Lie algebra $\mathfrak{g}$ is a linear space that is isomorphic to $\mathbb{R}^n$, where $n = \dim \mathcal{G}$, and we use the "hat" operator $^\wedge$ to map from $\mathbb{R}^n$ to the Lie algebra $\mathfrak{g}$ and the "vee" operator $\vee$ to map from $\mathfrak{g}$ to $\mathbb{R}^n$.

We can also map $\mathfrak{g} \to T_X\mathcal{G}$ for any $X \in \mathcal{G}$ simply by using left and right multiplication. That is, for $\xi \in \mathbb{R}^n$, both $X\xi^\wedge$ and $\xi^\wedge X \in T_X\mathcal{G}$ [29]. While both $X\xi^\wedge$ and $\xi^\wedge X$ result in elements in $T_X\mathcal{G}$, generally, $X\xi^\wedge \neq \xi^\wedge X$. These maps are also often used in differential equations when transporting tangent vectors to their correct tangent spaces. For example, if R is the rotation matrix from the body to the inertial frame and ω is the angular velocity of the body expressed in the body frame, then $\dot{R} = R\omega^\wedge$. Alternatively, if R is the rotation matrix from the inertial to the body frame, then $\dot{R} = -\omega^\wedge R$.

For matrix Lie groups, the exponential map at the identity $\exp_I : \mathfrak{g} \to \mathcal{G}$ is given by the usual matrix exponential, which is denoted by "exp" throughout this article [31]. Thus, we can map $\xi \in \mathbb{R}^n$ to a neighborhood of $I \in \mathcal{G}$ using $\exp(\xi^\wedge)$. We can also map any $\xi \in \mathbb{R}^n$ to the neighborhood of any $X \in \mathcal{G}$ by using $X\exp(\xi^\wedge)$ or $\exp(\xi^\wedge)X$, which again are different maps, as we discuss later.

Another useful operator is the adjoint of an element of $\mathcal{G}$, defined for matrix Lie groups as follows.

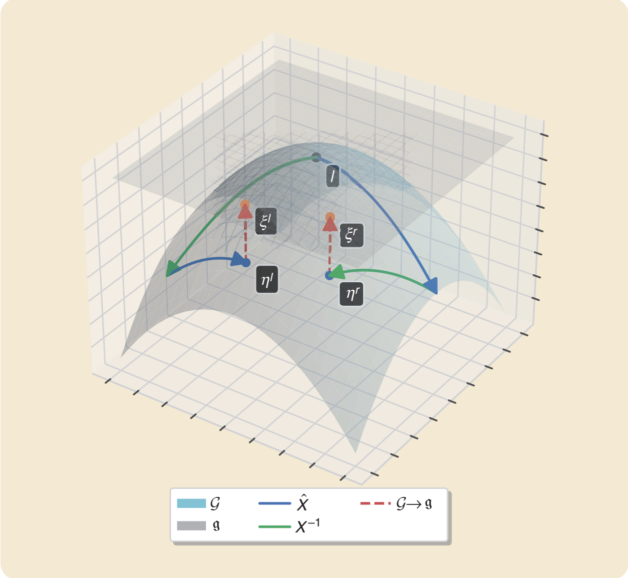

FIGURE 1 A comparison of right- and left-invariant errors on a model matrix Lie group. The state estimate $\hat{X}$ is shown in blue and the inverted true state X^{-1} in orange. The mapping of the right n^r and left n^l invariant errors in the Lie group $\mathcal{G}$ to ξ^r and ξ^l in the Lie algebra $\mathfrak{g}$ is displayed in red. Although the right and left errors are composed of the same transformations, the multiplication order results in different errors.

Definition 1 [31]

For any $X \in \mathcal{G}$, where $\mathcal{G}$ is a matrix Lie group, and for any $\xi \in \mathbb{R}^n$, where $\mathfrak{g}$ is the Lie algebra of $\mathcal{G}$, the adjoint map $\mathrm{Ad}_X : \mathfrak{g} \to \mathfrak{g}$ is defined by

$$\mathrm{Ad}_X(\xi^\wedge) = X\xi^\wedge X^{-1}. \tag{1}$$

Using the definition of the matrix exponential, it is straightforward to verify that

$$\exp(\mathrm{Ad}_X(\xi^\wedge)) = \exp(X\xi^\wedge X^{-1}) = X\exp(\xi^\wedge)X^{-1}. \tag{2}$$

For a fixed $X \in \mathcal{G}$, the adjoint map $\mathrm{Ad}_X()$ is linear. Its matrix representation $\mathrm{Ad}_X : \mathbb{R}^n \to \mathbb{R}^n$ operates directly on the Euclidean vectors and is defined through the relationship

$$(\mathrm{Ad}_X)\xi = (\mathrm{Ad}_X(\xi^\wedge))^\vee \tag{3}$$

where multiplying the Euclidean vector ξ by the matrix Ad_X is equivalent to the Lie algebra operation $\mathrm{Ad}_X(\xi^\wedge)$ mapped back to the equivalent Euclidean space. Whenever the adjoint is used in this article, it uses the matrix representation.

The adjoint matrix has a number of other useful properties, including [32]

$$\mathrm{Ad}_{X^{-1}} = \mathrm{Ad}_X^{-1} \tag{4}$$

$$\mathrm{Ad}_{XY} = \mathrm{Ad}_X\mathrm{Ad}_Y \tag{5}$$

$$X\exp(\xi^\wedge) = \exp((\mathrm{Ad}_X\xi)^\wedge)X \tag{6}$$

$$X\xi^\wedge = (\mathrm{Ad}_X\xi)^\wedge X \tag{7}$$

where the last two properties are derived by rearranging (1) and (2) and can be used to move a vector from left to right multiplication [33]. These properties will be used when converting between right- and left-invariant errors later in this article.

We can also define the "little ad" operator, which is useful when linearizing continuous process models, as follows.

Definition 2 [31]

If $\mathcal{G}$ is a matrix Lie group with Lie algebra $\mathfrak{g}$, then for any $\xi, \zeta \in \mathbb{R}^n$, we define the linear map

$$\mathrm{ad}_{\xi^\wedge}(\zeta^\wedge) = [\xi^\wedge, \zeta^\wedge] = \xi^\wedge\zeta^\wedge - \zeta^\wedge\xi^\wedge \tag{8}$$

where $[\cdot,\cdot]$ is called the Lie bracket. Similar to Ad, it is also often expressed by its matrix representation as $(\mathrm{ad}_{\xi^\wedge})\zeta \triangleq (\mathrm{ad}_{\xi^\wedge}(\zeta^\wedge))^\vee$.

The matrix representations of these two adjoint operators are related as follows.

Theorem 1 [31]
Given any $\xi \in R^n$,

$$\mathrm{Ad}_{\exp(\xi^\wedge)} = \exp(\mathrm{ad}_{\xi^\wedge}). \tag{9}$$

See "Example of Matrix Lie Group" for the application of these definitions to the matrix Lie groups $SO(2)$ and $SE(2)$.

FILTER BACKGROUND

In this section, we briefly review the EKF and the ESKF as a precursor to introducing the InEKF.

EKF

Given a state x defined by linear dynamics

$$\frac{d}{dt}x = Ax + Bu + w \qquad w \sim \mathcal{N}(0, Q\delta(t-\tau)) \tag{10}$$

and measurements given by the linear model

$$z = Hx + v \qquad v \sim \mathcal{N}(0, M) \tag{11}$$

the KF computes the optimal state estimate $\hat{x}$ of x given z in the mean-square-error sense [1]. Furthermore, the KF is globally asymptotically stable in a deterministic setting [2], meaning that it will converge to the true state given any initial estimate. It does this by propagating the mean and covariance of $\hat{x}$ through the dynamics in a prediction step and then correcting the mean and covariance of $\hat{x}$ in an update step.

Generally, most system dynamics and measurement models are nonlinear, so the KF cannot be used directly. The EKF circumvents this problem by linearizing both the system dynamics and measurement models using Jacobian linearization [34]. These steps are outlined in Figure 2(a).

Example of Matrix Lie Group

As an example of a matrix Lie group, consider the set of rotation matrices in two dimensions, that is, the special orthogonal group $SO(2)$, which can be defined as [33]

$$SO(2) = \{A \in M_2(\mathbb{R}) \mid A^T A = AA^T = I, \det(A) = 1\}. \tag{S3}$$

All the following hold for $A, B \in SO(2)$:

- *Closure*: AB is in $SO(2)$.
- *Associativity*: Matrix multiplication is associative.
- *Identity*: $I_2 \in SO(2)$.
- *Inverse*: $A^T = A^{-1} \in SO(2)$.

Thus, $SO(2)$ is a group with matrix multiplication as the operator.

Showing that $SO(2)$ is a smooth manifold is beyond the scope of this article, but we explain a few of the Lie group properties here. First, $SO(2)$ has dimension one; thus, $\mathfrak{so}(2)$, its Lie algebra, also has dimension one and can be defined as [33]

$$\mathfrak{so}(2) = \left\{ \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} \middle| \theta \in \mathbb{R} \right\} \tag{S4}$$

along with the operator

$$\xi \in \mathbb{R}^1, \quad \xi^\wedge = \begin{bmatrix} 0 & -\xi \\ \xi & 0 \end{bmatrix} \tag{S5}$$

and the matrix exponential

$$\exp(\xi^\wedge) = \begin{bmatrix} \cos(\xi) & -\sin(\xi) \\ \sin(\xi) & \cos(\xi) \end{bmatrix}. \tag{S6}$$

Due to the fact that matrices in $SO(2)$ and $\mathfrak{so}(2)$ commute, we have the following for $X \in SO(2), \xi \in \mathbb{R}^1$:

$$\begin{aligned} (Ad_X \xi)^\wedge = Ad_X(\xi^\wedge) &= X\xi^\wedge X^{-1} \\ &= XX^{-1}\xi^\wedge \\ &= \xi^\wedge \\ &\Rightarrow Ad_X = I. \end{aligned} \tag{S7}$$

Example S1

Using the adjoint, we can switch from left to right multiplication. Given $R \in SO(2), \xi \in \mathbb{R}$, we thus have

$$R\xi^\wedge = (Ad_R \xi)^\wedge R = \xi^\wedge R. \tag{S8}$$

Using Theorem 1 or the definition, we also arrive at

$$ad_{\xi^\wedge} = 0_{2\times 2}. \tag{S9}$$

Here, $SO(2)$ is shown embedded in $\mathbb{R}^2$ as the unit circle in Figure S1, along with its Lie algebra $\mathfrak{so}(2)$.

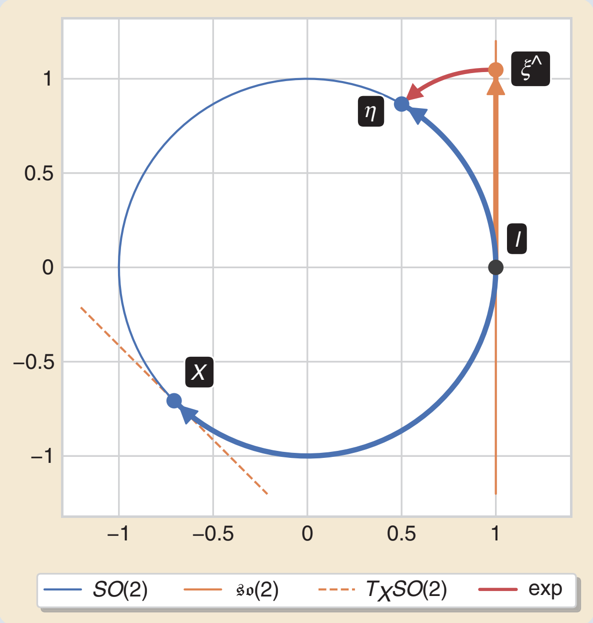

FIGURE S1 An illustration of $SO(2)$ embedded in $\mathbb{R}^2$. Included is an example tangent space $T_X SO(2)$, the Lie algebra $\mathfrak{so}(2)$, and a vector $\xi^\wedge \in \mathfrak{so}(2)$ with its exponential map to $\eta \in SO(2)$.

(Continued)

Example of Matrix Lie Group (*Continued*)

The special Euclidean group in two dimensions, or *SE*(2), is another matrix Lie group made up of a 2D rotation and translation, defined as

$$SE(2)=\left\{A\in M_3(\mathbb{R})\,\middle|\,A=\begin{bmatrix}R & p\\ 0_{1\times 3} & 1\end{bmatrix}, R\in SO(2), p\in\mathbb{R}^2\right\}. \quad \text{(S10)}$$

Note that *SE*(2) is also a group since $A,B\in SE(2)$ implies the following:

- *Closure*: $AB\in SE(2)$ since *SO*(2) is closed, and

$$AB=\begin{bmatrix}R_A & p_A\\ 0 & 1\end{bmatrix}\begin{bmatrix}R_B & p_B\\ 0 & 1\end{bmatrix}=\begin{bmatrix}R_AR_B & R_Ap_B+p_A\\ 0 & 1\end{bmatrix}. \quad \text{(S11)}$$

- *Associativity*: Matrix multiplication is associative.
- *Identity*: $I_3\in SE(2)$.
- *Inverse*: $A^{-1}=\begin{bmatrix}R_A^T & -R_A^Tp_A\\ 0 & 1\end{bmatrix}\in SO(2)$.

In addition, *SE*(2) has dimension three—one rotational dimension and two translational dimensions; thus, its Lie algebra $\mathfrak{se}(2)$ has dimension three and the form

$$\mathfrak{se}(2)=\left\{\begin{bmatrix}0 & -\xi_\theta & \xi_x\\ \xi\theta & 0 & \xi_y\\ 0 & 0 & 0\end{bmatrix}\,\middle|\,\begin{bmatrix}\xi_\theta\\ \xi_x\\ \xi_y\end{bmatrix}\in\mathbb{R}^3\right\}=\left\{\begin{bmatrix}\xi_\theta^\wedge & \xi_p\\ 0 & 0\end{bmatrix}\,\middle|\,\begin{bmatrix}\xi_\theta\\ \xi_p\end{bmatrix}\in\mathbb{R}^3\right\} \quad \text{(S12)}$$

where $\xi_\theta^\wedge$ is using the *SO*(2) hat operator. To define the adjoint, we first define the operator $\star$ for use in the following identity. For $a\in\mathbb{R}, b=[b_1\ \ b_2]^T\in\mathbb{R}^2$,

$$a^\wedge b=\begin{bmatrix}0 & -a\\ a & 0\end{bmatrix}\begin{bmatrix}b_1\\ b_2\end{bmatrix}=\begin{bmatrix}-b_2\\ b_1\end{bmatrix}a\triangleq b^\star a. \quad \text{(S13)}$$

Then, for $X\in SE(2)$ and $\xi\in\mathbb{R}^3$, since matrices commute in *SO*(2) and $\mathfrak{so}(2)$,

$$\begin{aligned}(Ad_X\xi)^\wedge=X\xi^\wedge X^{-1}&=\begin{bmatrix}R & p\\ 0 & 1\end{bmatrix}\begin{bmatrix}\xi_\theta^\wedge & \xi_\mathrm{p}\\ 0 & 0\end{bmatrix}\begin{bmatrix}R^\mathrm{T} & -R^\mathrm{T}p\\ 0 & 1\end{bmatrix}\\ &=\begin{bmatrix}R\xi_\theta^\wedge R^\mathrm{T} & -R\xi_\theta^\wedge R^\mathrm{T}p+R\xi_\mathrm{p}\\ 0 & 0\end{bmatrix}\\ &=\begin{bmatrix}\xi_\theta & -p^\star\xi_\theta+R\xi_\mathrm{p}\\ 0 & 0\end{bmatrix}=\left(\begin{bmatrix}1 & 0\\ -p^\star & R\end{bmatrix}\begin{bmatrix}\xi_\theta\\ \xi_\mathrm{p}\end{bmatrix}\right)^\wedge\\ &\Rightarrow Ad_X=\begin{bmatrix}1 & 0\\ -p^\star & R\end{bmatrix}.\end{aligned} \quad \text{(S14)}$$

Note that in some publications, ξ is defined with rotational states at the end rather than the start, that is, $\xi=[\xi_p^T\ \ \xi_\theta]^T$, which results in the columns and rows of the adjoint being flipped. Finally, for $\xi,\zeta\in\mathbb{R}^3$, since matrices in $\mathfrak{so}(2)$ commute,

$$\begin{aligned}(ad_{\xi^\wedge}\zeta)^\wedge=\xi^\wedge\zeta^\wedge-\zeta^\wedge\xi^\wedge&=\begin{bmatrix}\xi_\theta^\wedge\zeta_\theta^\wedge & \xi_\theta^\wedge\zeta_p\\ 0 & 0\end{bmatrix}-\begin{bmatrix}\zeta_\theta^\wedge\xi_\theta^\wedge & \zeta_\theta^\wedge\xi_p\\ 0 & 0\end{bmatrix}\\ &=\begin{bmatrix}0 & \xi_\theta^\wedge\zeta_p-\zeta_\theta^\wedge\xi_p\\ 0 & 0\end{bmatrix}=\left(\begin{bmatrix}0 & 0\\ -\xi_p^\star & \xi_\theta^\wedge\end{bmatrix}\zeta\right)^\wedge\\ &\Rightarrow ad_{\xi^\wedge}=\begin{bmatrix}0 & 0\\ -\xi_p^\star & \xi_\theta^\wedge\end{bmatrix}.\end{aligned} \quad \text{(S15)}$$

Table S1 provides a summary of properties for some of the common Lie groups used in robotics, and the readers should consult other works [32] for a more in-depth analysis.

TABLE S1 Various properties of typical Lie groups that are used in robotics.

$\mathcal{G}$	$\dim\mathcal{G}$	Matrix Size	$X\in\mathcal{G}$	Ad_X	$\xi\in\mathbb{R}^{\dim\mathcal{G}}$	$\xi^\wedge\in\mathfrak{g}$	$ad_{\xi^\wedge}$
SO(2)	1	2 × 2	R	I	$[\xi_\theta]$	$\begin{bmatrix}0 & -\xi_\theta\\ \xi_\theta & 0\end{bmatrix}$	0
SO(3)	3	3 × 3	R	R	$\begin{bmatrix}\xi_1\\ \xi_2\\ \xi_3\end{bmatrix}$	$\begin{bmatrix}0 & -\xi_3 & \xi_2\\ \xi_3 & 0 & -\xi_1\\ -\xi_2 & \xi_1 & 0\end{bmatrix}$	$\xi^\wedge$
SE(2)	3	3 × 3	$\begin{bmatrix}R & p\\ 0 & 1\end{bmatrix}$	$\begin{bmatrix}1 & 0\\ -p^* & R\end{bmatrix}$	$\begin{bmatrix}\xi_\theta\\ \xi_p\end{bmatrix}$	$\begin{bmatrix}\xi_\theta^\wedge & \xi_p\\ 0 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0\\ -\xi_p^* & \xi_\theta^\wedge\end{bmatrix}$
SE(3)	6	4 × 4	$\begin{bmatrix}R & p\\ 0 & 1\end{bmatrix}$	$\begin{bmatrix}R & 0\\ p^\wedge R & R\end{bmatrix}$	$\begin{bmatrix}\xi_\theta\\ \xi_p\end{bmatrix}$	$\begin{bmatrix}\xi_\theta^\wedge & \xi_p\\ 0 & 0\end{bmatrix}$	$\begin{bmatrix}\xi_\theta^\wedge & 0\\ \xi_p^\wedge & \xi_\theta^\wedge\end{bmatrix}$
$SE_2(2)$	5	4 × 4	$\begin{bmatrix}R & v & p\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}1 & 0 & 0\\ -v^* & R & 0\\ -p^* & 0 & R\end{bmatrix}$	$\begin{bmatrix}\xi_\theta\\ \xi_v\\ \xi_p\end{bmatrix}$	$\begin{bmatrix}\xi_\theta^\wedge & \xi_v & \xi_p\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0 & 0\\ -\xi_v^* & \xi_\theta^\wedge & 0\\ -\xi_p^* & 0 & \xi_\theta^\wedge\end{bmatrix}$
$SE_2(3)$	9	5 × 5	$\begin{bmatrix}R & v & p\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}R & 0 & 0\\ v^\wedge R & R & 0\\ p^\wedge R & 0 & R\end{bmatrix}$	$\begin{bmatrix}\xi_\theta\\ \xi_v\\ \xi_p\end{bmatrix}$	$\begin{bmatrix}\xi_\theta^\wedge & \xi_v & \xi_p\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}\xi_\theta^\wedge & 0 & 0\\ \xi_v^\wedge & \xi_\theta^\wedge & 0\\ \xi_p^\wedge & 0 & \xi_\theta^\wedge\end{bmatrix}$

Unfortunately, this linearization can introduce significant approximation errors in the filter, and it causes the EKF to lose many of the mathematical guarantees of the linear KF, except under strict assumptions [35]. A poor estimate often leads to a poor linearization point, which leads to another poor estimate and can begin a cycle of unacceptable performance. While an iterated KF [36] can solve for a poor linearization point, we later see with the InEKF that iteration is not necessary at all.

For an example of the EKF applied to visual–inertial navigation, see "Visual–Inertial Navigation Using the EKF, ESKF, and InEKF."

ESKF

The ESKF is an extension of the EKF, which seeks to track the covariance of the error rather than that of the predicted state [37]. The error state is defined as [38]

$$\delta x = x \ominus \hat{x} \tag{12}$$

where x is the true state, $\hat{x}$ is the state estimate, and the EKF is designed to track the error state δx. In a standard KF, $\ominus$ is generally defined using vector subtraction $x \ominus \hat{x} = x - \hat{x}$, but the ESKF allows a near-arbitrary choice of operation. Note that $\hat{x}$ is generally defined with deterministic dynamics, while x is stochastic.

Thus, in the prediction step, $\hat{x}$ is propagated forward with the incoming u, and the covariance of δx is updated using the Jacobian of the error dynamics [37], given by

$$A = \frac{\partial}{\partial \delta x}\left(\frac{d}{dt}\delta x\right)\Big|_{\hat{x},u} \triangleq \frac{\partial}{\partial \delta x} g_u(\delta x)\Big|_{\hat{x},u} \tag{13}$$

where g_u defines the error dynamics.

The update step is once again accomplished similarly to that of the standard EKF. We apply the standard EKF update step to δx, assuming that δx is reset to zero after each measurement step and is therefore zero throughout the prediction cycle. The result after the measurement update is the current estimate of the error that we can add to the state estimate using the inverse of the $\ominus$ operation,

$$\hat{x}_{+} = \hat{x} \oplus \delta x \tag{14}$$

and then δx is reset to zero.

The ESKF has a number of advantages over the standard EKF. One is that the error dynamics g are often much closer to linear than the actual dynamics f [37]. This allows for less approximation error in the Jacobian linearization, although the linearization still does depend on the current state estimate. The ESKF also allows for an arbitrary definition of error and, thus, an arbitrary choice of state estimate update $\oplus$, allowing for improved tracking on manifolds. Often, the state being tracked is not a vector space, and

```
$\hat{\Sigma}_x = \Sigma_0$;
$\hat{x} = x_0$;
if Between measurements then
    $\frac{\mathrm{d}}{\mathrm{d}t}\hat{x} = f(\hat{x}, u)$;
    $A = \frac{\partial f}{\partial x}|_{\hat{x},u}$;
    $G = -\frac{\partial f}{\partial u}|_{\hat{x},u}$;
    $\frac{\mathrm{d}}{\mathrm{d}t}\hat{\Sigma}_x = A\hat{\Sigma}_x + \hat{\Sigma}_x A^\top + GQG^\top$
if At measurement $z$ then
    $H = \frac{\partial h}{\partial x}|_{\hat{x}}$;
    $S = H\hat{\Sigma}_x H^\top + M$;
    $K = \hat{\Sigma}_x H^\top S^{-1}$;
    $\delta x = K(z - h(\hat{x}))$;
    $\hat{x} = \hat{x} + \delta x$;
    $\hat{\Sigma}_x = (I - KH)\hat{\Sigma}_x$
```

(a)

```
$\hat{\Sigma}_{\delta x} = \Sigma_0$;
$\hat{x} = x_0$;
if Between measurements then
    $\frac{\mathrm{d}}{\mathrm{d}t}\hat{x} = f(\hat{x}, u)$;
    $A = \frac{\partial g}{\partial \delta x}|_{\hat{x},u}$;
    $G = \frac{\partial g}{\partial \delta u}|_{\hat{x},u}$;
    $\frac{\mathrm{d}}{\mathrm{d}t}\hat{\Sigma}_{\delta x} = A\hat{\Sigma}_{\delta x} + \hat{\Sigma}_{\delta x} A^\top + GQG^\top$;
if At measurement $z$ then
    $H = \frac{\partial h}{\partial \delta x}|_{\hat{x}}$;
    $S = H\hat{\Sigma}_{\delta x} H^\top + M$;
    $K = \hat{\Sigma}_{\delta x} H^\top S^{-1}$;
    $\delta x = K(z - h(\hat{x}))$;
    $\hat{x} = \hat{x} \oplus \delta x$;
    $\hat{\Sigma}_{\delta x} = (I - KH)\hat{\Sigma}_{\delta x}$;
```

(b)

```
$\hat{\Sigma}^r = \Sigma_0$;
$\hat{X} = X_0$;
if Between measurements then
    $\frac{\mathrm{d}}{\mathrm{d}t}\hat{X} = f_u(\hat{X})$;
    $A = g_u(\exp(\xi^{r\wedge}))$;
    $\frac{\mathrm{d}}{\mathrm{d}t}\hat{\Sigma}^r = A\hat{\Sigma}^r + \hat{\Sigma}^r A^\top + Ad_{\hat{X}} Q Ad_{\hat{X}}^\top$;
if At measurement $z$ then
    $H\xi \triangleq \xi^\wedge b$;
    $S = H\hat{\Sigma}^r H^\top + \hat{R} M \hat{R}^\top$;
    $K = \hat{\Sigma}^r H^\top S^{-1}$;
    $\delta X = K(\Pi \hat{X} z - \Pi b)$;
    $\hat{X} = \exp(-\delta X)\hat{X}$;
    $\hat{\Sigma}^r = (I - KH)\hat{\Sigma}^r$;
```

(c)

FIGURE 2 Algorithms for the standard, error state, and right InEKFs: (a) algorithm 1: the EKF, (b) algorithm 2: the ESKF, and (c) algorithm 3: the right InEKF.

regular vector addition is ill-suited as an update operator, whereas something such as quaternion multiplication, in the case of rotational states, is a much better choice. These steps are summarized in Figure 2(b). We see in the next section how the InEKF leverages both of these advantages to their fullest in cases where the dynamics exhibit certain structural properties.

For an example of the ESKF applied to visual–inertial navigation, see "Visual–Inertial Navigation Using the EKF, ESKF, and InEKF."

InEKF

The InEKF uses a specially chosen error that gives the interaction among states, such as position and orientation, during the state estimate update through the matrix exponential.

Definition 3 [7]

The right- and left-invariant error between two elements $X, \hat{X} \in \mathcal{G}$ is given by

$$\eta^r \triangleq \hat{X}X^{-1} \quad \text{(right-invariant error)} \tag{15}$$

$$\eta^l \triangleq X^{-1}\hat{X} \quad \text{(left-invariant error)}. \tag{16}$$

Define the error states as $\xi^r \triangleq \log(\eta^r)^\vee \in \mathbb{R}^n$ for the right error and $\xi^l \triangleq \log(\eta^l)^\vee \in \mathbb{R}^n$ for the left error. Note that in some publications [39], the inverse definition is taken: $\eta^r = (\hat{X}X^{-1})^{-1} = X\hat{X}^{-1}$. We note that this alternate convention will simply reduce to a negative outside the logarithm above and, hence, a negative throughout the rest of the derivations.

Example 1

The right- and left-invariant errors received their names because they are invariant to right and left multiplications of the state, as follows given $Y \in \mathcal{G}$:

$$\eta^r = (\hat{X}Y)(XY)^{-1} = \hat{X}YY^{-1}X^{-1} = \hat{X}X^{-1} \tag{17}$$

$$\eta^l = (YX)^{-1}(Y\hat{X}) = X^{-1}Y^{-1}Y\hat{X} = X^{-1}\hat{X}. \tag{18}$$

The InEKF tracks the distributions of the error states ξ^r and ξ^l in the Lie algebra, allowing for more accurate uncertainty representation on group elements [40]. By solving for X in (15) and (16), in the update step, the state will be corrected as

$$X = \exp(-\xi^{r\wedge})\hat{X} \quad \text{(right-invariant update)} \tag{19}$$

$$X = \hat{X}\exp(-\xi^{l\wedge}) \quad \text{(left-invariant update)}. \tag{20}$$

These equations can also be viewed as defining a Gaussian on a Lie group [40]. For example, if $\xi^l, \xi^r \sim \mathcal{N}(0,\Sigma)$, then (19) defines a Gaussian on the Lie group $\mathcal{G}$, with $\hat{X}$ as the mean and covariance Σ, denoted $X \sim \mathcal{N}^r(\hat{X},\Sigma)$. Similarly, the left-invariant update (20) gives $X \sim \mathcal{N}^l(\hat{X},\Sigma)$.

Not only does this choice of error give improved state updates and uncertainty modeling, but when the system dynamics

$$\frac{d}{dt}X = f_u(X) \tag{21}$$

where u is the input, are restricted to the class of functions called *group affine*, we achieve state-independent log-linear error dynamics, as follows.

Theorem 2 [7]

If $\mathcal{G}$ is a Lie group, then the system in (21) is said to be group affine if $f_u(\cdot)$ satisfies

$$f_u(X \cdot Y) = f_u(X)\cdot Y + X \cdot f_u(Y) - X \cdot f_u(I) \cdot Y \tag{22}$$

for all $X, Y \in \mathcal{G}$. If this condition is satisfied, the right- and left-invariant error dynamics are trajectory independent, that is, independent of X and $\hat{X}$, and satisfy

$$\frac{d}{dt}\eta^r = g_u(\eta^r) \triangleq f_u(\eta^r) - \eta^r f_u(I) \tag{23}$$

$$\frac{d}{dt}\eta^l = g_u(\eta^l) \triangleq f_u(\eta^l) - f_u(I)\eta^l. \tag{24}$$

Example 2

In the case where the group operator $\cdot$ is a vector addition on $\mathbb{R}^n$, (22) simplifies to

$$f_u(X+Y) = (f_u(X)+Y) + (X+f_u(Y)) - (X+f_u(I)+Y) \tag{25}$$

$$= f_u(X) + f_u(Y) - f_u(0) \tag{26}$$

where 0 is the identity of $\mathbb{R}^n$. In this simple case, it can be seen that f_u must take the form of $f_u(X) = AX + b$, also known as an affine function. Hence, it can be helpful to think of group affine functions as affine functions simply defined on groups.

Furthermore, this theorem shows that the resulting error dynamics g are independent of the state X and state estimate $\hat{X}$ and dependent only on the error variable η, implying that any linearization will not be corrupted by poor state estimates.

If the group affine property is satisfied, the right or left error differential equation can then be linearized about the corresponding $\xi \in \mathbb{R}^n$ by defining A to satisfy [7]

$$g_u(\exp(\xi^\wedge)) \triangleq (A\xi)^\wedge + O(\|\xi\|^2). \tag{27}$$

This approximation is often done using a first-order approximation of the matrix exponential, $\eta \approx I + \xi^\wedge$. Using the above definition of A, the error dynamics satisfy

$$\frac{d}{dt}\xi = A\xi. \tag{28}$$

Equation (28) results in linearized error dynamics with second-order error. However, as one may expect from the definition of group affine, it turns out that this linearization introduces no approximation error. In other words, the above linearized dynamics of ξ perfectly capture the true dynamics of η, as stated in the following theorem.

Theorem 3 [7]

Consider the right- or left-invariant error η between any two trajectories that satisfy the group affine property. For arbitrary initial error $\xi_0 \in R^n$, if $\eta_0 = \exp(\xi_0^\wedge)$, then for all $t \geq 0$,

$$\eta_t = \exp(\xi_t^\wedge).$$

In other words, the nonlinear error η can be exactly recovered from the time-varying linear differential (28).

Thus, rather than attempting to track the Lie group error η, which is likely highly nonlinear, we can instead track its representation in the vector space Lie algebra ξ, which has linear dynamics.

The InEKF therefore exploits the benefits of the ESKF to its fullest by using an error that provides an improved state update step through the matrix exponential and using specific structural constraints on the process model that result in state-independent log-linear error dynamics.

Example 3

Consider the system on $SO(3)$ given by dynamics $(d/dt)R = f_\omega(R) \triangleq R\omega^\wedge$, with $\omega \in \mathbb{R}$ a known angular velocity. These dynamics are group affine since

$$\begin{aligned} f_\omega(RS) &= RS\omega^\wedge \\ &= R\omega^\wedge S + RS\omega^\wedge - R\omega^\wedge S \\ &= f_\omega(R)S + Rf_\omega(S) - Rf_\omega(I)S \end{aligned} \tag{29}$$

and thus have right and left error dynamics:

$$\begin{aligned} \frac{d}{dt}\eta^r &= g_u(\eta^r) \triangleq f_u(\eta^r) - \eta^r f_u(I) \\ &= \eta^r\omega^\wedge - \eta^r\omega^\wedge = 0, \\ \frac{d}{dt}\eta^l &= g_u(\eta^l) \triangleq f_u(\eta^l) - f_u(I)\eta^l \\ &= \eta^l\omega^\wedge - \omega^\wedge\eta^l \\ &\approx (I + \xi^{l\wedge})\omega^\wedge - \omega^\wedge(I + \xi^{l\wedge}) \\ &= \xi^{l\wedge}\omega^\wedge - \omega^\wedge\xi^{l\wedge} \\ &= -(\mathrm{ad}_\omega\xi^l)^\wedge. \end{aligned}$$

Thus, the dynamics of η^r and η^l can be exactly tracked by using the linear dynamics given by $(d/dt)\xi^r = 0$ and $(d/dt)\xi^l = -\mathrm{ad}_\omega\xi^l$.

The InEKF allows for many of the same guarantees as the standard linear KF that are rarely seen for nonlinear systems; in particular, it is a proven asymptotically stable observer [7, Th. 4]. The steps of the right InEKF are summarized in Figure 2(c), with details described in the following sections. For an example of the InEKF applied to visual–inertial navigation, see "Visual–Inertial Navigation Using the EKF, ESKF, and InEKF."

While there are several articles in the literature that compare the InEKF to the EKF or ESKF, there is limited discussion or analysis of the difference between the right and left InEKFs [8], [26], [27], [28]. While both variants will generally outperform other observers for a given robotic application, knowing which variant to use for a given application is nuanced and can affect the performance of the filter. In the following section, we clearly explain how to use the InEKF as well as the differences between the left and right InEKFs.

RIGHT AND LEFT ERROR

In robotics, the Lie groups $SE(2)$, $SE(3)$, and their variants are often of interest. In these scenarios, an element $X \in G$ represents a transformation from a local frame to the global frame, which we denote as X_l^g. This state will have a rotation of the local frame to the global frame R_l^g and position p_{gl}^g from the global to the local frame represented in the global frame. In this sense, the right-invariant error can be seen as the error in the global frame, while the left-invariant error is the error in the local frame.

For a simple example, consider the true transformation from the local to the global frame, $X_l^g \in SE(3)$, of the form

$$X_l^g = \begin{bmatrix} R_l^g & p_{gl}^g \\ 0 & 1 \end{bmatrix} \tag{30}$$

while $\hat{X}_l^g$ is the estimate of X_l^g of the form

$$\hat{X}_l^g = \begin{bmatrix} \hat{R}_l^g & \hat{p}_{gl}^g \\ 0 & 1 \end{bmatrix}. \tag{31}$$

The right- and left-invariant errors will thus be

$$\begin{aligned} \eta^r = \hat{X}_l^g (X_l^g)^{-1} &= \begin{bmatrix} \hat{R}_l^g & \hat{p}_{gl}^g \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_l^{gT} & -R_l^{gT} p_{gl}^g \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \hat{R}_l^g R_l^{gT} & \hat{p}_{gl}^g - \hat{R}_l^g R_l^{gT} p_{gl}^g \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \hat{R}_l^g R_g^\ell & \hat{p}_{gl}^g - \hat{R}_l^g R_g^\ell p_{gl}^g \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{32}$$

$$\begin{aligned} \eta^\ell = (X_l^g)^{-1} \hat{X}_l^g &= \begin{bmatrix} R_l^{gT} & -R_l^{gT} p_{gl}^g \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{R}_l^g & \hat{p}_{gl}^g \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_l^{gT} \hat{R}_l^g & R_l^{gT}(\hat{p}_{gl}^g - p_{gl}^g) \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_g^\ell \hat{R}_l^g & R_g^\ell(\hat{p}_{gl}^g - p_{gl}^g) \\ 0 & 1 \end{bmatrix}. \end{aligned} \tag{33}$$

Note that η^r has its orientation and position errors in the global frame, while η^ℓ is in the local frame. Throughout the rest of the article, we use the shorthand X, R, and p for X_l^g, R_l^g, and p_{gl}^g, respectively.

Furthermore, the distributions in (19) and (20) also follow this paradigm, where (19) represents noise applied

Visual–Inertial Navigation Using the EKF, ESKF, and InEKF

In this sidebar, we show how the three filters listed in Figure 2 might be defined for the common problem of visual–inertial navigation. We assume that the system equations of motion are given by

$$\begin{aligned}\dot{R}^i_b &= R^i_b(\omega^b - w_\omega)^\wedge\\ \dot{v}^i &= g^i + R^i_b(a^b - w_a)\\ \dot{p}^i &= v^i - R^i_b w_v\end{aligned} \tag{S16}$$

where $p^i \in \mathbb{R}^3$ is the position of a robot expressed in the inertial frame, $v^i \in \mathbb{R}^3$ is the velocity, $g^i \in \mathbb{R}^3$ is a constant vector expressing the acceleration of gravity expressed in the inertial frame, and $R^i_b \in SO(3)$ is the rotation matrix from body to inertial frames. Here, the inputs to the system are given by the measured body frame acceleration $a^b \in \mathbb{R}^3$, which is the actual body frame acceleration plus noise $w_a \sim \mathcal{N}(0, \Sigma_a)$, and the measured body frame angular velocity $\omega^b \in \mathbb{R}^3$, which is the actual body frame angular velocity plus noise $w_\omega \sim \mathcal{N}(0, \Sigma_\omega)$. We have also added process noise $w_v \sim \mathcal{N}(0, \Sigma_v)$ to the evolution of the position to account for kinematic error and expressed this noise in the body frame to be consistent with the other noise terms. For exposition purposes, we assume that the robot makes a single range and bearing measurement (for example, using a lidar) of a known landmark $L^i \in \mathbf{R}^3$, where the measurement is given by

$$z = R^{i\top}_b(L^i - p^i) + w_z \tag{S17}$$

where the measurement noise is $w_z \sim \mathcal{N}(0, M)$.

ESTIMATION USING A NAIVE EKF

To estimate the state using a naive EKF that does not exploit the group structure of $SO(3)$, the attitude can be parameterized using the 3–2–1 Euler angles ϕ, θ, and ψ, denoting the roll, pitch, and yaw angles, respectively [S2]. Defining $\Theta = (\phi, \theta, \psi)^\top$, the estimated state in Figure 2(a) is defined as $\hat{x} \triangleq (\hat{\Theta}^\top, \hat{v}^{i\top}, \hat{p}^{i\top})^\top$, and the input is $u = (\omega^{b\top}, a^{b\top})^\top$, and the estimated equations of motion are

$$\begin{aligned}\dot{\hat{\Theta}} &= S(\hat{\Theta})\omega^b\\ \dot{\hat{v}}^i &= g^i + R(\hat{\Theta})a^b\\ \dot{\hat{p}}^i &= \hat{v}^i\end{aligned} \tag{S18}$$

where

$$R(\Theta) \triangleq \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi\\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi\\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix},$$

$$S(\Theta) \triangleq \begin{pmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta\\ 0 & c_\phi & -s_\phi\\ 0 & s_\phi \sec_\theta & c_\phi \sec_\theta \end{pmatrix} \tag{S19}$$

and $c_* = \cos(*)$, $s_* = \sin(*)$, $t_* = \tan(*)$, and $\sec_* = \sec(*)$. The Jacobians on lines 5 and 6 in Figure 2(a) are given by

$$A = \left.\frac{\partial f}{\partial x}\right|_{\hat{x},u} = \begin{pmatrix} \dfrac{\partial S(\hat{\Theta})\omega^b}{\partial \hat{\Theta}} & 0_{3\times3} & 0_{3\times3}\\ \dfrac{\partial R(\hat{\Theta})a^b}{\partial \hat{\Theta}} & 0_{3\times3} & 0_{3\times3}\\ 0_{3\times3} & I_{3\times3} & 0_{3\times3} \end{pmatrix},$$

$$G = -\left.\frac{\partial f}{\partial u}\right|_{\hat{x},u} = \begin{pmatrix} -S(\hat{\Theta}) & 0_{3\times3} & 0_{3\times3}\\ 0_{3\times3} & -R(\hat{\Theta}) & 0_{3\times3}\\ 0_{3\times3} & 0_{3\times3} & -R(\hat{\Theta}) \end{pmatrix} \tag{S20}$$

where it is clear that A and G are nonlinear and depend strongly on the estimated state $\hat{x}$.

The predicted output is given by

$$\hat{z} = h(\hat{x}) = R(\hat{\Theta})^\top (L^i - \hat{p}^i) \tag{S21}$$

and therefore, the Jacobian on line 9 in Figure 2(a) is

$$H = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}} = \begin{pmatrix} \dfrac{\partial R(\hat{\Theta})^\top (L^i - \hat{p}^i)}{\partial \hat{\Theta}} & 0_{3\times3} & -R(\hat{\Theta}) \end{pmatrix} \tag{S22}$$

which is again highly nonlinear and strongly dependent on the estimated state $\hat{x}$. Therefore, both the covariance prediction step on line 7 in Figure 2(a) and the covariance update step on line 14 in Figure 2(a) are nonlinear and state dependent due to the strong state dependence of A, G, and H.

ESTIMATION USING AN ESKF

The ESKF when used with attitude estimation is sometimes called the multiplicative EKF [S3], [S4], [S5] or indirect EKF [S6]. In the application considered in this sidebar, the state is defined as the tuple $x = \{R^i_b, v^i, p^i\}$, and the state estimate as the tuple $\hat{x} = \{\hat{R}^i_b, \hat{v}^i, \hat{p}^i\}$ is propagated according to the nonlinear equations

$$\begin{aligned}\dot{\hat{R}}^i_b &= \hat{R}^i_b \omega^{b\wedge}\\ \dot{\hat{v}}^i &= g^i + \hat{R}^i_b a^b\\ \dot{\hat{p}}^i &= \hat{v}^i.\end{aligned} \tag{S23}$$

The typical definition of the error state in position and velocity is computed as

$$\xi^i_p = \hat{p}^i - p^i, \quad \xi^i_v = \hat{v}^i - v^i \tag{S24}$$

and the attitude error state is computed as either

$$\xi^i_{\theta_r} = \log(\hat{R}^i_b R^{i\top}_b)^\vee \quad \text{or} \quad \xi^b_{\theta_\ell} = \log(R^{i\top}_b \hat{R}^i_b)^\vee \tag{S25}$$

where $\xi^i_{\theta_r} \in \mathbb{R}^3$ is the right attitude error and $\xi^b_{\theta_\ell} \in \mathbb{R}^3$ is the left attitude error, and note that the right error is an attitude deviation expressed in the inertial frame, and the left error is an attitude deviation expressed in the body frame.

The evolution of the error states is different depending on whether the right or left attitude error is used. In the case of right attitude error, we have that

$$\begin{aligned}\frac{d}{dt}\xi^i_p &= \dot{\hat{p}}^i - \dot{p}^i = \hat{v}^i - v^i + R^i_b w_v \\ &= \xi^i_v + \exp(-\xi^{i\wedge}_{\theta_r})\hat{R}^i_b w_v \\ &\approx \xi^i_v + (I - \xi^{i\wedge}_{\theta_r})\hat{R}^i_b w_v \approx \xi^i_v + \hat{R}^i_b w_v. \end{aligned} \tag{S26}$$

For the velocity, we have

$$\begin{aligned}\frac{d}{dt}\xi^i_v &= \dot{\hat{v}}^i - \dot{v}^i \\ &= g^i + \hat{R}^i_b a^b - (g^i + R^i_b(a^b - w_a)) \\ &= (\hat{R}^i_b - R^i_b)a^b + R^i_b w_a \\ &= (I - \exp(-\xi^{i\wedge}_{\theta_r}))\hat{R}^i_b a^b + \exp(-\xi^{i\wedge}_{\theta_r})\hat{R}^i_b w_a \\ &\approx (I - (I - \xi^{i\wedge}_{\theta_r}))\hat{R}^i_b a^b + \hat{R}^i_b w_a \\ &= \xi^{i\wedge}_{\theta_r}\hat{R}^i_b a^b + \hat{R}^i_b w_a \\ &= -(\hat{R}^i_b a^b)^\wedge \xi^i_{\theta_r} + \hat{R}^i_b w_a. \end{aligned} \tag{S27}$$

For the evolution of the attitude error, we use the fact that

$$\frac{d}{dt}\exp(\xi^{i\wedge}_{\theta_r}) \approx \frac{d}{dt}(I + \xi^{i\wedge}_{\theta_r}) = \frac{d}{dt}\xi^{i\wedge}_{\theta_r} \tag{S28}$$

to get

$$\begin{aligned}\frac{d}{dt}\xi^{i\wedge}_{\theta_r} &= \dot{\hat{R}}^i_b R^{i\top}_b + \hat{R}^i_b \frac{d}{dt}(R^{i\top}_b) \\ &= \hat{R}^i_b \omega^{b\wedge} R^{i\top}_b - \hat{R}^i_b R^{i\top}_b R^i_b(\omega^b - w_\omega)^\wedge R^{i\top}_b \\ &= (\hat{R}^i_b w_\omega)^\wedge \hat{R}^i_b R^{i\top}_b \approx (\hat{R}^i_b w_\omega)^\wedge (I + \xi^{i\wedge}_{\theta_r}) \approx (\hat{R}^i_b w_\omega)^\wedge. \end{aligned} \tag{S29}$$

Therefore, the error state satisfies

$$\frac{d}{dt}\begin{pmatrix} \xi_{\theta r}{}^i \\ \xi^i_v \\ \xi^i_p \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ -(\hat{R}^i_b a^b)^\wedge & 0 & 0 \\ 0 & I & 0 \end{pmatrix}\begin{pmatrix} \xi^i_p \\ \xi^i_v \\ \xi_{\theta r} \end{pmatrix} + \begin{pmatrix} \hat{R}^i_b & 0 & 0 \\ 0 & \hat{R}^i_b & 0 \\ 0 & 0 & \hat{R}^i_b \end{pmatrix}\begin{pmatrix} w_\omega \\ w_a \\ w_v \end{pmatrix}. \tag{S30}$$

We see that the A and G matrices on lines 5 and 6 in Figure 2(b) depend on the estimated state $\hat{x}$ through $\hat{R}^i_b$ and on the input u via a^b.

In the case of left attitude error, we again get that $(d/dt)\xi^i_p = \xi^i_v + \hat{R}^i_b w_v$, but following a similar derivation as above, we get that $(d/dt)\xi^i_v = -\hat{R}^i_b a^{b\wedge}\xi_{\theta\ell} + \hat{R}^i_b w_a$. Similarly, we get that the left attitude error satisfies $(d/dt)\xi^b_{\theta_\ell} = -\omega^{b\wedge}\xi^b_{\theta_\ell} + w_\omega$. Therefore, the error state satisfies

$$\frac{d}{dt}\begin{pmatrix} \xi_{\theta\ell}{}^b \\ \xi^i_v \\ \xi_p{}^i \end{pmatrix} = \begin{pmatrix} -\omega^{b\wedge} & 0 & 0 \\ -\hat{R}^i_b a^{b\wedge} & 0 & \\ 0 & I & 0 \end{pmatrix}\begin{pmatrix} \xi_{\theta\ell} \\ \xi^i_v \\ \xi^i_p \end{pmatrix} + \begin{pmatrix} I & 0 & 0 \\ 0 & \hat{R}^i_b & 0 \\ 0 & 0 & \hat{R}^i_b \end{pmatrix}\begin{pmatrix} w_\omega \\ w_a \\ w_v \end{pmatrix} \tag{S31}$$

and we see that the A and G matrices on lines 5 and 6 in Figure 2(b) for this case depend on the estimated state $\hat{x}$ through $\hat{R}^i_b$ and on the input u via a^b and ω^b.

For the output equation we have that

$$\begin{aligned} z &= h(x) = R^{i\top}_b(L^i - p^i) = R^{i\top}_b(L^i - p^i + \hat{p}^i - \hat{p}^i) \\ &= R^{i\top}_b(L^i - \hat{p}^i + \xi^i_p). \end{aligned} \tag{S32}$$

For the right error, where $\hat{R}^i_b R^{i\top}_b = \exp(\xi^{i\wedge}_{\theta_r}) \approx I + \xi^{i\wedge}_{\theta_r}$ implies that $R^{i\top}_b \approx \hat{R}^{i\top}_b(I + \xi^{i\wedge}_{\theta_r})$, we have that

$$\begin{aligned} z &\approx \hat{R}^{i\top}_b(I + \xi^{i\wedge}_{\theta_r})((L^i - \hat{p}^i) + \xi^i_p) \\ &= \hat{R}^{i\top}_b(L^i - \hat{p}^i) + \hat{R}^{i\top}_b\xi^i_p + \hat{R}^{i\top}_b\xi^{i\wedge}_{\theta_r}(L^i - \hat{p}^i) + \hat{R}^{i\top}_b\xi^{i\wedge}_{\theta_r}\xi^i_p \\ &\approx \hat{R}^{i\top}_b(L^i - \hat{p}^i) + (-\hat{R}^{i\top}_b(L^i - \hat{p}^i)^\wedge \quad 0_{3\times3} \quad \hat{R}^{i\top}_b)\begin{pmatrix} \xi^i_{\theta_r} \\ \xi^i_v \\ \xi^i_p \end{pmatrix} \\ &\triangleq \hat{z} + H\delta x \end{aligned} \tag{S33}$$

where we see that the Jacobian H on line 9 in Figure 2(b) depends on the estimated state $\hat{x}$ through $\hat{R}^i_b$ and $\hat{p}^i$.

Similarly, for the left error, we can derive

$$z \approx \hat{z} + ((\hat{R}^{i\top}_b(L^i - \hat{p}^i))^\wedge \quad 0_{3\times3} \quad \hat{R}^{i\top}_b)\begin{pmatrix} \xi^b_{\theta_\ell} \\ \xi^i_v \\ \xi^i_p \end{pmatrix} \tag{S34}$$

where again the Jacobian H on line 9 in Figure 2(b) depends on the estimated state $\hat{x}$ through $\hat{R}^i_b$ and $\hat{p}^i$.

ESTIMATION USING THE InEKF

We now show the equivalent formulations using the InEKF. The right InEKF defines the state estimate as

$$\hat{X} = \begin{pmatrix} \hat{R}^{\hat{\imath}}_b & \hat{v}^{\hat{\imath}} & \hat{p}^{\hat{\imath}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{S35}$$

and, therefore, the error using

$$\begin{aligned} \eta^r = \hat{X}X^{-1} &= \begin{pmatrix} \hat{R}^{\hat{\imath}}_b & \hat{v}^{\hat{\imath}} & \hat{p}^{\hat{\imath}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} R^i_b & v^i & p^i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \\ &= \begin{pmatrix} \hat{R}^{\hat{\imath}}_b R^{i\top}_b & \hat{v}^{\hat{\imath}} - \hat{R}^{\hat{\imath}}_b R^{i\top}_b v^i & \hat{p}^{\hat{\imath}} - \hat{R}^{\hat{\imath}}_b R^{i\top}_b p^i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned} \tag{S36}$$

where we have used the superscript i to denote the true inertial frame and $\hat{\imath}$ to denote the estimated inertial frame. Therefore, for the right InEKF, the error state is given by, using the approximation $\eta = I + \xi^\wedge$,

$$\xi_{\theta r} = \log(\hat{R}^{\hat{\imath}}_b R^{i\top}_b)^\vee \tag{S37}$$

$$\xi^{\hat{\imath}}_v \approx \hat{v}^{\hat{\imath}} - \hat{R}^{\hat{\imath}}_b R^{i\top}_b v^i \tag{S38}$$

$$\xi^{\hat{\imath}}_p \approx \hat{p}^{\hat{\imath}} - \hat{R}^{\hat{\imath}}_b R^{i\top}_b p^i \tag{S39}$$

where it is clear that the errors are defined relative to the estimated inertial frame and that the subtractions are carried out in the correct frame, as opposed to the simpler but naive position and velocity errors defined in (S24). Interestingly enough, the right attitude error is defined identically to (S25), but it is now clear that the rotation error is from the inertial frame to the estimated inertial frame.

(*Continued*)

Visual–Inertial Navigation Using the EKF, ESKF, and InEKF (*Continued*)

Differentiating the position and velocity errors gives

$$
\begin{aligned}
\frac{d}{dt}\xi_p^{\hat{\imath}} &= \dot{\hat{p}}^{\hat{\imath}} - \dot{\hat{R}}_b^{\hat{\imath}} R_b^{i\top} p^i - \hat{R}_b^{\hat{\imath}} \frac{d}{dt}(R_b^{i\top}) p^i - \hat{R}_b^{\hat{\imath}} R_b^{i\top} \dot{p}^i \\
&= \hat{v}^{\hat{\imath}} - \hat{R}_b^{\hat{\imath}} \omega^{i\wedge} R_b^{i\top} p^i + \hat{R}_b^{\hat{\imath}} R_b^{i\top} R_b^i \omega^{i\wedge} R_b^{i\top} p^i \\
&\quad - \hat{R}_b^{\hat{\imath}} R_b^{i\top} R_b^i w_\omega^\wedge R_b^{i\top} p^i - \hat{R}_b^{\hat{\imath}} R_b^{i\top} (v^i - R_b^i w_v) \\
&= \hat{v}^{\hat{\imath}} - \hat{R}_b^{\hat{\imath}} R^{i\top} v^i - (\hat{R}_b^{\hat{\imath}} w_\omega)^\wedge (\hat{p}^{\hat{\imath}} - \delta p^i) + \hat{R}_b^{\hat{\imath}} w_v \\
&\approx \xi_v^{\hat{\imath}} + \hat{p}^{\hat{\imath}\wedge} \hat{R}_b^{\hat{\imath}} w_\omega + \hat{R}_b^{\hat{\imath}} w_v \\
\frac{d}{dt}\xi_v^{\hat{\imath}} &\approx g^{i\wedge} \xi_{\theta r} + \hat{R}_b^{\hat{\imath}} w_a + \hat{v}^{\hat{\imath}\wedge} \hat{R}_b^{\hat{\imath}} w_\omega
\end{aligned} \tag{S40}
$$

and we have shown in the previous section that $\dot{\xi}_{\theta_r} = \hat{R}_b^i w_\omega$. Therefore, the error state satisfies

$$
\begin{pmatrix} \dot{\xi}_{\theta_r} \\ \dot{\xi}_v^{\hat{\imath}} \\ \dot{\xi}_v^{\hat{\imath}} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & g^{i\wedge} \\ 0 & I & 0 \end{pmatrix} \begin{pmatrix} \xi_{\theta r} \\ \xi_v^{\hat{\imath}} \\ \xi_p^{\hat{\imath}} \end{pmatrix} + \begin{pmatrix} \hat{R}_b^{\hat{\imath}} & 0 & 0 \\ \hat{v}^{\hat{\imath}\wedge} \hat{R}_b^{\hat{\imath}} & \hat{R}_b^{\hat{\imath}} & 0 \\ \hat{p}^{\hat{\imath}\wedge} \hat{R}_b^{\hat{\imath}} & 0 & \hat{R}_b^{\hat{\imath}} \end{pmatrix} \begin{pmatrix} w_\omega \\ w_a \\ w_v \end{pmatrix} \tag{S41}
$$

where the matrix multiplying the noise is seen from Table S1 to be $\mathrm{Ad}_{\hat{X}}$, and we see that in this case, the Jacobian A on line 5 in Figure 2(c) is constant but that $\mathrm{Ad}_{\hat{X}}$ is state dependent, which implies a state-dependent propagation of the error covariance $\hat{\Sigma}^r$.

For the measurement model, we have that

$$
\begin{aligned}
z^r = R_b^{i\top}(L^i - p^i) + w_z &= \begin{pmatrix} R_b^{i\top} & -R_b^{i\top} v^i & -R_b^{i\top} p^i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} L^i \\ 0 \\ 1 \end{pmatrix} + \tilde{w}_z \\
&= X^{-1} b + \tilde{w}_z
\end{aligned} \tag{S42}
$$

which implies from (53) that it is a right-invariant measurement model. Therefore, from (58), we have that the right innovation is given by

$$
\begin{aligned}
V^r &= \xi^{r\wedge} b + \hat{X}\tilde{w}_z \\
&= \begin{pmatrix} \xi_{\theta_r}^\wedge & \xi_v^{\hat{\imath}} & \xi_p^{\hat{\imath}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} L^i \\ 0 \\ 1 \end{pmatrix} + \hat{X}\tilde{w}_z \\
&= \begin{pmatrix} \xi_{\theta_r}^\wedge L^i + \xi_p^{\hat{\imath}} \\ 0 \\ 1 \end{pmatrix} + \hat{X}\tilde{w}_z .
\end{aligned} \tag{S43}
$$

Letting $\Pi \triangleq (I_{3\times 3} \;\; 0_{3\times 1} \;\; 0_{3\times 1})$ gives

$$
\begin{aligned}
\Pi V^r &= \xi_{\theta_r}^\wedge L^i + \xi_p^{\hat{\imath}} + \hat{R}_b^i w_z \\
&= (-L^{i\wedge} \;\; 0_{3\times 3} \;\; I_{3\times 3}) \begin{pmatrix} \xi_{\theta_r} \\ \xi_v^{\hat{\imath}} \\ \xi_p^{\hat{\imath}} \end{pmatrix} + \hat{R}_b^i w_z \\
&\triangleq -H^r \xi^r + \hat{R} w_z
\end{aligned} \tag{S44}
$$

which defines H^r and Π on lines 8 and 11 in Figure 2(c).

The left InEKF defines the left error using

$$
\begin{aligned}
\eta_\ell = X^{-1}\hat{X} &= \begin{pmatrix} R_b^i & v^i & p^i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} \hat{R}_{\hat{b}}^i & \hat{v}^i & \hat{p}^i \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} R_b^{i\top} \hat{R}_{\hat{b}}^i & R_b^{i\top}(\hat{v}^i - v^i) & R_b^{i\top}(\hat{p}^i - p^i) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.
\end{aligned} \tag{S45}
$$

Therefore, for the left InEKF, the error state is given by, using the approximation $\eta = I + \xi^\wedge$,

$$
\begin{aligned}
\xi_{\theta\ell} &= \log(R_b^{i\top} \hat{R}_{\hat{b}}^i)^\vee \\
\xi_v^b &\approx R_b^{i\top}(\hat{v}^i - v^i) \\
\xi_p^b &\approx R_b^{i\top}(\hat{p}^i - p^i)
\end{aligned} \tag{S46}
$$

where we see that in contrast to the position and velocity errors in (S24), (S38), and (S39), the position and velocity errors are defined in the body frame, and the left-invariant attitude error is similar to the left error in (S25), but it is clear that the rotation error is between the estimated body frame and the body frame. Differentiating the position and velocity errors gives, after some algebra,

$$
\begin{aligned}
\frac{d}{dt}\xi_p^b &\approx -\omega^{b\wedge}\xi_p^b + \xi_v^b + w_v \\
\frac{d}{dt}\xi_v^b &\approx -\omega^{b\wedge}\xi_v^b - a^{b\wedge}\xi_{\theta\ell} + w_a .
\end{aligned} \tag{S47}
$$

We have shown in the previous section that $(d/dt)\xi_{\theta\ell} = \omega^{b\wedge}\xi_{\theta\ell} + w_\omega$. Therefore, for the left-invariant error, the error state satisfies

$$
\frac{d}{dt}\begin{pmatrix} \xi_{\theta\ell} \\ \xi_v^b \\ \xi_p^b \end{pmatrix} = \begin{pmatrix} -\omega^{b\wedge} & 0 & 0 \\ -a^{b\wedge} & -\omega^{b\wedge} & 0 \\ 0 & I & -\omega^{b\wedge} \end{pmatrix} \begin{pmatrix} \xi_{\theta\ell} \\ \xi_v^b \\ \xi_p^b \end{pmatrix} + \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} w_\omega \\ w_a \\ w_v \end{pmatrix} \tag{S48}
$$

and we see that in this case, the Jacobian A on line 5 in Figure 2(c) is independent of the state $\hat{X}$ but depends on the input u, and the error covariance propagation on line 6 in Figure 2(c) will be independent of the state.

For the measurement model, from (73), we get that

$$
\begin{aligned}
H^\ell &= H^r Ad_{\hat{X}} \\
&= (L^{i\wedge} \;\; 0_{3\times 3} \;\; -I_{3\times 3}) \begin{pmatrix} \hat{R}_b^{\hat{\imath}} & 0 & 0 \\ \hat{v}^{\hat{\imath}\wedge} \hat{R}_b^{\hat{\imath}} & \hat{R}_b^{\hat{\imath}} & 0 \\ \hat{p}^{\hat{\imath}\wedge} \hat{R}_b^{\hat{\imath}} & 0 & \hat{R}_b^{\hat{\imath}} \end{pmatrix} \\
&= ((L^i - \hat{p}^{\hat{\imath}})^\wedge \hat{R}_b^{\hat{\imath}} \;\; 0_{3\times 3} \;\; -\hat{R}_b^{\hat{\imath}})
\end{aligned} \tag{S49}
$$

which defines H^ℓ on line 8 in Figure 2(c).

REFERENCES

[S2] M. D. Shuster, "A survey of attitude representations," *J. Astronautical Sci.*, vol. 41, no. 4, pp. 439–517, 1993.

[S3] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *AIAA J. Guidance, Control Dyn.*, vol. 5, no. 5, pp. 417–429, 1982.

[S4] P. Martin and E. Salaun, "Generalized multiplicative extended Kalman filter for aided attitude and heading reference system," in *Proc. AIAA Guidance, Navigation Control Conf.*, p. 8300, 2010, doi: 10.2514/6.2010-8300.

[S5] D. P. Koch, D. O. Wheeler, R. W. Beard, T. W. McLain, and K. M. Brink, "Relative multiplicative extended Kalman filter for observable GPS-denied navigation," *Int. J. Robotic Res.*, vol. 39, no. 9, pp. 1085–1121, 2020, doi: 10.1177/0278364920903094.

[S6] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation a tutorial for quaternion algebra," Univ. Minnesota, Tech. Rep. 2005-002, Rev. 57, Mar. 2005.

in the global frame and (20) represents noise applied in the local frame. When specifying an initial mean and covariance for the InEKF, these are the distributions that are initialized.

One must take care because even with the same mean and covariance, $\mathcal{N}^r$ and $\mathcal{N}^\ell$ represent entirely different distributions. This can be seen in Figure 3, which shows various distributions in $SE(2)$ with identical covariances of $\Sigma = \text{diag}(.5°, .01, .01)$ and varying means. The left distribution results in an elliptical distribution, as one would expect from a Gaussian with a diagonal covariance; however, the right distribution has a very different shape. When initializing a right InEKF, it must be taken into account that an initial diagonal covariance *is not* an elliptical Gaussian but instead results in a banana-shaped distribution because the error is applied before transforming by $\hat{X}$.

There is an exact linear conversion between ξ^ℓ and ξ^r that allows us to switch between the two and their covariances and that intuitively involves the adjoint. Its derivation is as follows [18]:

$$\begin{aligned}
\eta^r = \hat{X}X^{-1} \quad &= \hat{X}X^{-1}\hat{X}\hat{X}^{-1} = \hat{X}\eta^\ell\hat{X}^{-1} \\
\Rightarrow \exp(\xi^{r\wedge}) \quad &= \hat{X}\exp(\xi^{\ell\wedge})\hat{X}^{-1} \\
&= \exp(\hat{X}\xi^{\ell\wedge}\hat{X}^{-1}) \\
&= \exp((\text{Ad}_{\hat{X}}\xi^\ell)^\wedge\hat{X}\hat{X}^{-1}) \\
\Rightarrow \xi^r \quad &= \text{Ad}_{\hat{X}}\xi^\ell. \qquad (34)
\end{aligned}$$

A near-identical derivation results in

$$\xi^\ell = \text{Ad}_{\hat{X}^{-1}}\xi^r. \qquad (35)$$

Thus, to initialize a right InEKF with an elliptical Gaussian, the following conversion should be applied:

$$\Sigma^r = \text{Ad}_{\hat{X}}\Sigma^\ell\text{Ad}_{\hat{X}}^\top \qquad (36)$$

where Σ^ℓ is a diagonal matrix.

It should also be noted that it may be preferable to track the global frame X_g^l rather than the local frame X_l^g in some circumstances. In these scenarios, it is simple to see that the above conventions are inverted; the right-invariant error can be seen as the error in the local frame, while the left-invariant error is in the global frame. Further, many of the results in the rest of the article will also be inverted. For conciseness and simplicity, we focus on tracking the local frame X_l^g in this article, as the derivations involving X_g^l follow similarly.

PROCESS MODEL

To gain insight into the linearized dynamics A^r and A^ℓ, we introduce the notion of left- and right-invariant dynamics.

Invariant Dynamics

Definition 4 [41]

The system $\dot{z} = f_u(z)$ is said to be left invariant if f_u satisfies

$$f_u(XY) = Xf_u(Y) \qquad (37)$$

for all $X, Y \in \mathcal{G}$. It is right invariant if f_u satisfies

$$f_u(XY) = f_u(X)Y. \qquad (38)$$

Example 4

The rotational dynamics $(d/dt)R_\ell^g = R_\ell^g\omega^{\ell\wedge}$, where ω^ℓ is the angular velocity expressed in the local frame, are left invariant. Alternatively, the rotational dynamics $(d/dt)R_\ell^g = -\omega^{g\wedge}R_\ell^g$, where ω^g is the angular velocity expressed in the global frame, are right invariant.

Since left multiplication does not change the dynamics of left-invariant systems, it follows that, generally, left-invariant systems must evolve through right multiplication, which modifies the local frame of X. Similarly, a right-invariant system will modify the global frame. Notice from the definition of group affine in Theorem 2 that if a system is right or left invariant, it is also group affine, but the converse is not necessarily true.

Right- and left-invariant systems result in simple-to-linearize process models. From the definition of g_u in (23) and (24), if a system is right invariant, the linearized left error dynamics A^ℓ will be 0, and if a system is left invariant, linearized right error dynamics A^r will be 0. For many robotic

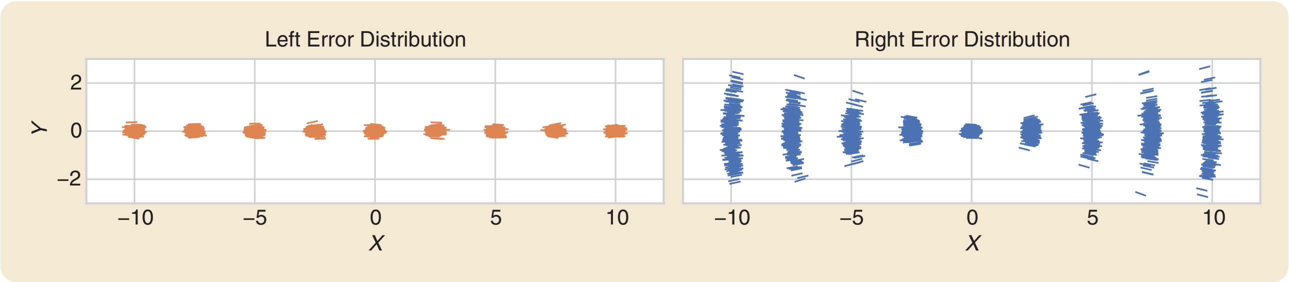

FIGURE 3 A comparison of right and left Lie group distributions on $SE(2)$. Each plot shows nine distributions, each with $\Sigma = \text{diag}(.5°, .01, .01)$, and means with varying x values. We sample 250 group elements from each distribution, with each $X \in SE(2)$ position shown on the x- and y-axes and rotation shown as the orientation of each line. Notice how different the right and left distributions are, even with identical mean and covariance. The (a) left error distribution and (b) right error distribution.

systems, angular velocity or linear acceleration is measured in the local frame, and therefore, the dynamics are left invariant or close to it. In these cases, A^r is either 0 or constant [7], [18], [20] and we focus on these cases throughout the article.

Intuitively, it makes sense that there is little transforming of the right error, which tracks in the global frame, needed between time steps in a left-invariant system since changes are occurring to the local frame. On the other hand, the left error, tracking in the local frame, will likely need to be rotated or transformed according to the most recent control u since the local frame is what is evolving. Thus, A^ℓ, while not dependent on the state, is often a function of the controls u [7].

Stochastic Models

Thus far, we have dealt entirely with deterministic models. For left-invariant systems, we introduce process noise into the system as

$$\frac{d}{dt}X = f_u(X) + Xw^\wedge, \qquad w \sim \mathcal{N}(0, Q\delta(t-\tau)) \tag{39}$$

where $w^\wedge$ is transferred to the tangent space of X. In this case, we can interpret the process noise as being added in the local frame. For right-invariant systems, process noise can be added in the global frame by using the term $w^\wedge X$ instead of $Xw^\wedge$ in (39) and with the following results interchanging right for left.

Introducing the process noise into the right and left error dynamics of (23) and (24) results in [7]

$$\frac{d}{dt}\eta^r = g_u(\eta^r) - \mathrm{Ad}_{\hat{X}} w^\wedge \eta^r \tag{40}$$

$$\frac{d}{dt}\eta^\ell = g_u(\eta^\ell) - w^\wedge \eta^\ell. \tag{41}$$

Following a similar linearization as in (28) and neglecting terms of order $O(\|\xi\|\|w\|)$, the stochastic dynamics of ξ^r and ξ^ℓ are [7]

$$\frac{d}{dt}\xi^r = A^r\xi + \mathrm{Ad}_{\hat{X}} w \tag{42}$$

$$\frac{d}{dt}\xi^\ell = A^\ell(u)\xi + w \tag{43}$$

and consequently, the covariance prediction equations are

$$\frac{d}{dt}\hat{\Sigma}^r = A^r\hat{\Sigma}^r + \hat{\Sigma}^r A^{rT} + \mathrm{Ad}_{\hat{X}} Q \mathrm{Ad}_{\hat{X}}^\top \tag{44}$$

$$\frac{d}{dt}\hat{\Sigma}^\ell = A^\ell(u)\hat{\Sigma}^\ell + \hat{\Sigma}^\ell A^{\ell\top}(u) + Q. \tag{45}$$

Since the right InEKF is tracking error in the global frame, it intuitively makes sense that the locally applied noise covariance Q must be transformed to the global frame using $\mathrm{Ad}_{\hat{X}}$, as in (36).

This presents a nuanced decision to be made. Given a left-invariant system, the covariance prediction step will be dependent on either $\mathrm{Ad}_{\hat{X}}$ in a right filter or on u through A^ℓ in a left filter. In practice, error in the conversion of Q seems to have less impact on the performance of the filter than the use of inaccuracies in A^ℓ. Furthermore, if the filter is running correctly, errors in $\mathrm{Ad}_{\hat{X}}$ should ideally converge to zero, whereas noise in u, and, thus, in $A^\ell(u)$, will likely remain close to the same magnitude throughout. This implies that the right InEKF is the preferred choice for most left-invariant or almost left-invariant systems, but the type of measurement plays a much larger role when choosing a filter, as covered later in the "Measurement Model" section.

Discretization

The linearized dynamics can also be used in discrete time as follows, assuming a constant A^ℓ and A_r over Δt [18], [42]:

$$\Phi^r = \exp(A^r\Delta t) \qquad \xi^r_{t+1} = \Phi^r\xi^r_t + \mathrm{Ad}_{\hat{X}} w\Delta t \tag{46}$$

$$\Phi^\ell = \exp(A^\ell\Delta t) \qquad \xi^\ell_{t+1} = \Phi^\ell\xi^\ell_t + w\Delta t. \tag{47}$$

The resulting covariance prediction equations will be

$$\hat{\Sigma}^r_+ = \Phi^r\hat{\Sigma}^r\Phi^{rT} + \mathrm{Ad}_{\hat{X}} Q \mathrm{Ad}_{\hat{X}}^\top(\Delta t)^2 \tag{48}$$

$$\hat{\Sigma}^\ell_+ = \Phi^\ell\hat{\Sigma}^\ell\Phi^{\ell\top} + Q(\Delta t)^2. \tag{49}$$

Example 5

Continuing with the process model $f_\omega(R) = R\omega^\wedge$ and its right linearization from Example 3 and remembering that the adjoint in $SO(3)$ is R, the discrete covariance prediction equations are

$$\begin{aligned} &\Phi^r = \exp(A^r\Delta t) = \exp(0\Delta t) = I \\ \Rightarrow\ &\check{\Sigma}^r = \hat{\Sigma}^r + RQR^\top(\Delta t)^2 \\ &\Phi^l = \exp(A^l\Delta t) = \exp(-\mathrm{ad}_\omega\Delta t) = \exp(-\mathrm{ad}_{\omega\Delta t}) = \mathrm{Ad}_{\exp(\omega\Delta t)} \\ \Rightarrow\ &\check{\Sigma}^l = \mathrm{Ad}_{\exp(\omega\Delta t)}\hat{\Sigma}^l\mathrm{Ad}^\top_{\exp(\omega\Delta t)} + Q(\Delta t)^2. \end{aligned} \tag{50}$$

A conversion between Φ^r and Φ^ℓ using (34) and (35) can be derived as follows [18]:

$$\begin{aligned} \xi^r_{t+1} &= \mathrm{Ad}_{\hat{X}_{t+1}}\xi^\ell_{t+1} = \mathrm{Ad}_{\hat{X}_{t+1}}\Phi^\ell\xi^\ell_t \\ &= \mathrm{Ad}_{\hat{X}_{t+1}}\Phi^\ell\mathrm{Ad}_{\hat{X}_t^{-1}}\xi^r_t \\ \Rightarrow \Phi^r &= \mathrm{Ad}_{\hat{X}_{t+1}}\Phi^\ell\mathrm{Ad}_{\hat{X}_t^{-1}}. \end{aligned} \tag{51}$$

Similarly,

$$\Phi^\ell = \mathrm{Ad}_{\hat{X}_{t+1}^{-1}}\Phi^r\mathrm{Ad}_{\hat{X}_t} \tag{52}$$

allowing for easier derivations and computations of either.

Imperfect InEKF

In some scenarios, it may be the case that not all the desired states to track fit into a Lie group, such as wheel radii, camera intrinsics, and so on. While the InEKF cannot be used in these cases, an "imperfect" InEKF [9] can be used by using an additional error ζ alongside η. An imperfect

InEKF loses the guarantees of the InEKF but still generally outperforms other state-of-the-art methods [18], [20].

The imperfect InEKF is used by deriving the error dynamics by hand for both η and ζ rather than leaning upon (23) and (24) for g and then using the approximation $\eta = \exp(\xi^{\wedge}) \approx I + \xi^{\wedge}$ to linearize. Measurement models are derived similarly by appending the necessary additional columns onto the linearized measurement model H. An example of this can be seen in the later imperfect InEKF example.

MEASUREMENT MODEL

While any measurement model and corresponding innovation that can be linearized about ξ^r or ξ^ℓ can be used, in practice, there are a few models that linearize particularly easily and with no dependence on the current state.

Invariant Measurement Models

Definition 5

The right- and left-invariant measurements from a state X are given by

$$z^r = X^{-1}b + \tilde{w}_m \quad (\text{right} - \text{invariant measurement}) \tag{53}$$

$$z^\ell = Xb + \tilde{w}_m \quad (\text{left} - \text{invariant measurement}) \tag{54}$$

where $\tilde{w}_m = [w_m \;\; 0]^\top$, with $w_m \sim \mathcal{N}(0, M)$, and b is a constant vector.

As opposed to the frame conventions with the invariant errors, the right measurement is a measurement of some linear combination of the state in the local frame, while the left measurement is in the global frame.

Example 6

Consider a 2D GPS measurement. Since it is a global measurement, it is a left measurement with the following b, given $X \in SE(2)$:

$$z^\ell = Xb + \tilde{w}_m = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} w_m \\ 0 \end{bmatrix} = \begin{bmatrix} p + w_m \\ 1 \end{bmatrix}. \tag{55}$$

Definition 6

Along with these measurement models, the following innovations are defined:

$$V^r = \hat{X}(z^r - \hat{z}^r) \quad (\text{right} - \text{invariant innovation}) \tag{56}$$

$$V^\ell = \hat{X}^{-1}(z^\ell - \hat{z}^\ell) \quad (\text{left} - \text{invariant innovation}) \tag{57}$$

with z as the actual measurement and $\hat{z}$ as the expected measurement using the state estimate.

For the special Euclidean groups, the bottom rows of V are often identically zero and are truncated using an auxiliary matrix $\Pi = [I \;\; 0]$. For the special orthogonal groups, we simply have $\Pi = I$. These measurement models and innovations are used because it allows for the following first-order linearization, with the approximation $\eta = \exp(\xi) \approx I + \xi^{\wedge}$ [8]:

$$\begin{aligned} V^r &= \hat{X}(z^r - \hat{z}^r) \\ &= \hat{X}(X^{-1}b + \tilde{w}_m - \hat{X}^{-1}b) \\ &= \eta^r b + \hat{X}\tilde{w}_m - b \\ &\approx (I + \xi^{r\wedge})b + \hat{X}\tilde{w}_m - b \\ &= \xi^{r\wedge}b + \hat{X}\tilde{w}_m \\ \Rightarrow \Pi V^r &= \Pi\xi^{r\wedge}b + \Pi\hat{X}w_m \triangleq H^r\xi^r + \Pi\hat{X}w_m. \end{aligned} \tag{58}$$

A near-identical linearization exists for V^ℓ about ξ^ℓ, as follows:

$$V^\ell = \hat{X}^{-1}(z^\ell - \hat{z}^\ell) \tag{59}$$

$$= \hat{X}^{-1}Xb + \hat{X}^{-1}b - b + \hat{X}^{-1}\tilde{w}_m \tag{60}$$

$$= (\eta^\ell)^{-1}b + \hat{X}^{-1}b - b + \hat{X}^{-1}\tilde{w}_m \tag{61}$$

$$\approx (I - \xi^{\ell\wedge})b + \hat{X}^{-1}b - b + \hat{X}^{-1}\tilde{w}_m \tag{62}$$

$$= -\xi^{\ell\wedge}b + \hat{X}^{-1}b + \hat{X}^{-1}\tilde{w}_m \tag{63}$$

$$\Rightarrow \Pi V^\ell = -\Pi\xi^{\ell\wedge}b + \Pi\hat{X}^{-1}w_m \triangleq H^\ell\xi^\ell + \Pi\hat{X}^{-1}w_m. \tag{64}$$

This linearization is not dependent on the state, and given a constant b, it results in a constant H. Succinct formulas for defining H from the right- and left-invariant innovations are thus

$$H^r\xi^r \triangleq \Pi\xi^{r\wedge}b, \qquad H^\ell\xi^\ell \triangleq -\Pi\xi^{\ell\wedge}b. \tag{65}$$

Additionally, the invariant innovation covariances are

$$\operatorname{Cov}(\Pi V^r) = S^r = H^r\Sigma^r H^{rT} + \Pi\hat{X}M\hat{X}^\top\Pi^T \tag{66}$$

$$\operatorname{Cov}(\Pi V^\ell) = S^\ell = H^\ell\Sigma^\ell H^{\ell\top} + \Pi\hat{X}^{-1}M\hat{X}^{-\top}\Pi^T. \tag{67}$$

The innovation also has a simplified version, for computational purposes, as follows:

$$\begin{aligned} v^r &= \Pi V^r = \Pi\hat{X}(z^r - \hat{z}^r) = \Pi\hat{X}(z^r - \hat{X}^{-1}b) \\ &= \Pi\hat{X}z^r - \Pi b \end{aligned} \tag{68}$$

where oftentimes $\Pi b = 0$ and can be omitted. A similar simplification exists for the left innovation, as follows:

$$\begin{aligned} v^\ell &= \Pi V^\ell = \Pi\hat{X}^{-1}(z^\ell - \hat{z}^\ell) = \Pi\hat{X}^{-1}(z^\ell - \hat{X}b) \\ &= \Pi\hat{X}^{-1}z^\ell - \Pi b. \end{aligned} \tag{69}$$

Other Measurement Models

As mentioned above, any measurement model that can be linearized about ξ_r or ξ_ℓ may be used.

Example 7

Consider a full pose measurement $z = X\exp(-w_m^{\wedge})$, with $X \in SE(2), w_m \sim \mathcal{N}(0_3, M)$. Using $\log\colon SE(2) \to \mathfrak{se}(2)$ and defining $^{\vee}$ as the inverse of $^{\wedge}$, one possible left innovation is

$$V^\ell = \log(z^{-1}\hat{z})^{\vee} = \log(\exp(w_m^{\wedge})X^{-1}\hat{X})^{\vee} \tag{70}$$

$$= \log(\exp(w_m^{\wedge})\eta^\ell)^{\vee} \tag{71}$$

$$\approx \log(\exp(w_m^{\wedge} + \xi^{\ell\wedge}))^{\vee} = w_m + \xi^\ell \tag{72}$$

where combining the exponentials is done using a first-order approximation of the Baker–Campbell–Hausdorff equation [31]. This makes the linearized measurement model $H^{\ell} = I$. Since the measurement is the full pose, it can also be formulated as a right measurement.

Converting Measurements

Furthermore, a measurement linearized about the right error can still be used in a left filter and vice versa. This can be done using the conversion shown in (34) and (35), as follows:

$$H^{r}\xi^{r} = (H^{r}\mathrm{Ad}_{\hat{X}})\xi^{\ell} \triangleq H^{\ell}\xi^{\ell} \tag{73}$$

$$H^{\ell}\xi^{\ell} = (H^{\ell}\mathrm{Ad}_{\hat{X}^{-1}})\xi^{r} \triangleq H^{r}\xi^{r}. \tag{74}$$

This conversion does introduce a dependence on the current state estimate and should be avoided when possible. This conversion is also equivalent to converting the state covariance to the alternate error, performing the appropriate alternate update step, and then converting the covariance back to the original error. We find the conversion of the measurement Jacobians to be more straightforward and to have a smaller reliance on terms canceling out numerically.

Due to the dependence on the state estimate introduced when converting measurements, when a single measurement is being used in the filter, the filter should match the measurement type. However, when both a right and a left measurement are to be used, preference should be given to the more accurate, frequent, or critical measurement or the process model differences discussed previously.

SELECTING A FILTER

As a brief overview of the prior sections, a rough outline of the ingredients and decision-making process of using the InEKF follows:

1) Confirm that the process model is group affine, or alternatively, use the imperfect InEKF or another InEKF formulation.
2) Determine whether the measurements are right or left invariant. Right-invariant measurements measure in the body frame, and left-invariant measurements measure in the global frame. If they are not right or left invariant, find a way to linearize about the right- or left-invariant errors.
3) Based on the type of measurement being used, choose a right- or left-invariant error. Choose an error type that best matches the most frequent or critical measurements. If there is no preference, go with the right-invariant error.

Note that alternate linearization methods may include standard Jacobian linearization, dropping higher-order terms, other approximations, and so on. This decision-making process has been summarized in Figure 4, and the differences between the two filters are outlined in Table 1.

EXAMPLES

To illustrate how these principles might be used, we present a number of simple examples step by step to show how to derive the filter. We note that these examples are purposefully straightforward in order to illustrate the basic principles of the InEKF. In addition, see "InEKF Open Source Library" for our open source C++ and Python library and the code for the following examples.

InEKF Example

We step through the derivation of both the right and left InEKFs on $SE(2)$ for a simple odometry motion model with GPS, landmark, and compass sensors. We track orientation R and position $p = [p_x \ \ p_y]^{\top}$. The state $X \in SE(2)$ and the error state $\xi \in \mathbb{R}^3$ are

$$X = \begin{bmatrix} R & p \\ 0_{1\times 2} & 1 \end{bmatrix}, \quad \xi^{\wedge} = \begin{bmatrix} \xi_{\theta} \\ \xi_{x} \\ \xi_{y} \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -\xi_{\theta} & \xi_{x} \\ \xi_{\theta} & 0 & \xi_{y} \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathrm{Ad}_{X}\xi = \begin{bmatrix} 1 & 0 \\ -p^{\star} & R \end{bmatrix}\xi, \quad \text{with } p^{\star} = \begin{bmatrix} p_x \\ p_y \end{bmatrix}^{\star} = \begin{bmatrix} -p_y \\ p_x \end{bmatrix} \tag{75}$$

where ξ_{θ}, ξ_{x}, and ξ_{y} are respectively the θ, x, and y invariant errors on the Lie algebra.

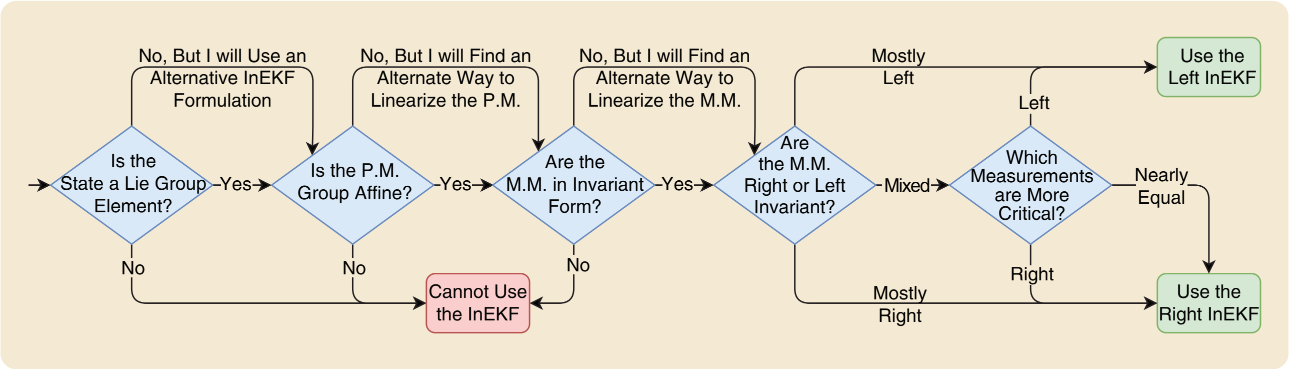
No, But I will Use an Alternative InEKF Formulation
No, But I will Find an Alternate Way to Linearize the P.M.
No, But I will Find an Alternate Way to Linearize the M.M.
Mostly Left
Use the Left InEKF
Left
Is the State a Lie Group Element?
Yes
Is the P.M. Group Affine?
Yes
Are the M.M. in Invariant Form?
Yes
Are the M.M. Right or Left Invariant?
Mixed
Which Measurements are More Critical?
Nearly Equal
No
No
No
Cannot Use the InEKF
Right
Mostly Right
Use the Right InEKF

FIGURE 4 The decision-making process when using the InEKF. Alternate linearization methods may include standard Jacobian linearization, dropping higher-order terms, other approximations, and so on. P.M.: process model; M.M.: measurement model.

TABLE 1 A comparison of right- and left-invariant errors, measurements, and KFs. Differences are highlighted in red. Generally, the left versions are considered tracking error in the local frame, while right errors are in the global frame. Elements after a prediction are noted with a ˇ and after an update with a ˆ

	Left	Right
Error	$\exp(\xi^{l\wedge}) = \eta^l = X^{-1}\hat{X}$	$\exp(\xi^{r\wedge}) = \eta^r = \hat{X}X^{-1}$
Distribution	$\hat{X} = X\exp(\xi^{l\wedge}) \sim \mathcal{N}_l(X, \Sigma^l)$	$\hat{X} = \exp(\xi^{r\wedge})X \sim \mathcal{N}_r(X, \Sigma^r)$
Conversion	$\xi^l = \mathrm{Ad}_{\hat{X}^{-1}}\xi^r$	$\xi^r = \mathrm{Ad}_{\hat{X}}\xi^l$
Predict Step	$\frac{d}{dt}\hat{X} = f_u(\hat{X})$ $\frac{d}{dt}\hat{\Sigma}^l = A^l\hat{\Sigma}^l + \hat{\Sigma}^l A^{lT} + Q$	$\frac{d}{dt}\hat{X} = f_u(\hat{X})$ $\frac{d}{dt}\hat{\Sigma}^r = A^r\hat{\Sigma}^r + \hat{\Sigma}^r A^{rT} + \mathrm{Ad}_{\hat{X}}Q\mathrm{Ad}_{\hat{X}}^T$
Predict Dependence (if Dynamics Are Left Invariant)	A^l is likely a function of u.	A^r is likely constant, and $\mathrm{Ad}_{\hat{X}}$ is a function of $\hat{X}$.
Measurement Model	$z^l = Xb + w$	$z^r = X^{-1}b + w$
Linearized Measurement Model	$H^l\xi^l = -\Pi\xi^{l\wedge}b$	$H^r\xi^r = \Pi\xi^{r\wedge}b$
Innovation	$v^l = \Pi\check{X}^{-1}z - \Pi b$	$v^r = \Pi\check{X}z - \Pi b$
Innovation Covariance	$S = H^l\check{\Sigma}^l H^{lT} + \hat{R}^T M\hat{R}$	$S = H^r\check{\Sigma}^r H^{rT} + \hat{R}M\hat{R}^T$
Update Step	$K = \check{\Sigma}^l H^{lT} S^{-1}$ $\hat{X} = \check{X}\exp(-Kv^l)$ $\hat{\Sigma}^l = (I - KH^l)\check{\Sigma}^l$	$K = \check{\Sigma}^r H^{rT} S^{-1}$ $\hat{X} = \exp(-Kv^r)\check{X}$ $\hat{\Sigma}^r = (I - KH^r)\check{\Sigma}^r$

Process Model

We begin with a discrete-time process model, as is often common when integrating odometry measurements. The process model is

$$X_{t+1} = X_t U_t \exp(w_t^{\wedge}), \qquad w_t \sim \mathcal{N}(0, \Sigma) \tag{76}$$

with $U_t \in SE(2)$ as the change in odometry between time step t and $t+1$. Note that (76) is the discretization, up to first order, of the continuous-time model

$$\begin{aligned} \frac{d}{dt}X &= f_u(X) + Xw^{\wedge} \\ &= Xu^{\wedge} + Xw^{\wedge}, \quad w \sim \mathcal{N}(0, \Sigma\delta(t-\tau)) \end{aligned} \tag{77}$$

with $U_t = \exp(u^{\wedge}\Delta t)$. Note that the following holds for $X, Y \in SE(2)$:

$$\begin{aligned} f_u(XY) &= XY(u)^{\wedge} \\ &= Xu^{\wedge}Y + XYu^{\wedge} - Xu^{\wedge}Y \\ &= f_u(X)Y + Xf_u(Y) - Xf_u(I)Y. \end{aligned} \tag{78}$$

Thus, f satisfies the group affine property. Next, we define and linearize g for both the right and left filters using Theorem 2 and the first-order approximation $\eta = \exp(\xi) \approx I + \xi^{\wedge}$:

$$\begin{aligned} g_u(\eta^r) &= f_u(\eta^r) - \eta^r f_u(I) = \eta^r u - \eta^r u = 0 \\ &= (0\xi^r)^{\wedge} \triangleq (A^r\xi^r)^{\wedge} \end{aligned} \tag{79}$$

$$\begin{aligned} g_u(\eta^{\ell}) &= f_u(\eta^{\ell}) - f_u(I)\eta^{\ell} = \eta^{\ell}u - u\eta^{\ell} \\ &\approx (I + \xi^{\ell\wedge})u - u(I + \xi^{\ell\wedge}) \\ &= \xi^{\ell\wedge}u - u\xi^{\ell\wedge} = (-\mathrm{ad}_u\xi^{\ell})^{\wedge} \triangleq (A^{\ell}\xi^{\ell})^{\wedge}. \end{aligned} \tag{80}$$

InEKF Open Source Library

To allow for easy implementation, we release an open source C++ InEKF library with an additional Python interface. It is built with modularity in mind, allowing for easy inheritance and extension, in both Python and C++, of the base process and measurement models. This makes implementing the InEKF for many process and measurement models simple and straightforward to do. Furthermore, all Lie groups that are generally used in robotics are included and heavily templated to allow for an arbitrary number of additional columns and Euclidean states.

Included are a number of examples, including an InEKF for underwater navigation [20] and an InEKF–SLAM implementation using the Victoria Park dataset. The library can be found online [S7].

In addition, we release all the code to run the simulations found in this article as open source, also available online [S8]. The figures were all generated using the library's Python bindings.

REFERENCES

[S7] E. Potokar, "Invariant extended Kalman filter library," Bitbucket. 2024. Accessed: Oct. 1, 2024. [Online]. Available: https://bitbucket.org/frostlab/inekf/

[S8] E. Potokar, "Invariant extended Kalman filter tutorial code," Bitbucket. 2024. Accessed: Oct. 1, 2024. [Online]. Available: https://bitbucket.org/frostlab/inekf_tutorial/

Thus, we have $A^r = 0$, $A^\ell = -\mathrm{ad}_u$, and when applied in discrete time and using Theorem 1, we have [18]

$$\Phi^r = \exp(0\Delta t) = I \tag{81}$$

$$\begin{aligned}\Phi^\ell &= \exp(-\mathrm{ad}_u \Delta t) = \exp(\mathrm{ad}_{-u\Delta t}) \\ &= \mathrm{Ad}_{\exp(-u\Delta t)} = \mathrm{Ad}_{U_t^{-1}}.\end{aligned} \tag{82}$$

Discretization removes the need to calculate u from U, which fits the paradigm we have mentioned, where for left-invariant systems, right-invariant filters generally have a constant A^r and Φ^r, while the left-invariant filter's A^ℓ and Φ^ℓ are dependent upon the controls.

GPS Measurements

The GPS measurement model is

$$z_p = p + w_p, \qquad w_p \sim N(0_2, \Sigma_p) \tag{83}$$

which can be rewritten as a left-invariant measurement:

$$\begin{bmatrix} z_p \\ 1 \end{bmatrix} = \begin{bmatrix} R & p \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} 0_{2\times 1} \\ 1 \end{bmatrix} + \begin{bmatrix} w_p \\ 0 \end{bmatrix} \triangleq Xb_p + \tilde{w}_p. \tag{84}$$

Thus, using the formula in (65),

$$H_p \xi^\ell = -\Pi \xi^{\ell\wedge} b_p = -\Pi \begin{bmatrix} 0 & -\xi_\theta^\ell & \xi_x^\ell \\ \xi_\theta^\ell & 0 & \xi_y^\ell \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \xi^\ell. \tag{85}$$

As discussed in previous sections, although linearized about ξ^ℓ, we can still use the measurement model with a right filter and ξ^r by the conversion in (34), although it does introduce a dependence on the current state estimate.

Landmark Measurements

A measurement of the relative position of a landmark with respect to a robot's frame follows a similar linearization process. If the known landmark's position in the world frame is $L = [L_x, L_y]^\top$, then the measurement is

$$z_L = R^\top (L - p) + w_L, \qquad w_L \sim \mathcal{N}(0_2, \Sigma_L) \tag{86}$$

which can be put into right-invariant form as

$$\begin{bmatrix} z_L \\ 1 \end{bmatrix} = \begin{bmatrix} R^\top & -R^\top p \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} L \\ 1 \end{bmatrix} + \begin{bmatrix} w_L \\ 0 \end{bmatrix} \triangleq X^{-1} b_L + \tilde{w}_L \tag{87}$$

and, thus, linearized as

$$\begin{aligned} H_L \xi^r = \Pi \xi^{r\wedge} b_L &= \Pi \begin{bmatrix} 0 & -\xi_\theta^r & \xi_x^r \\ \xi_\theta^r & 0 & \xi_y^r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} L_x \\ L_y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} -L_y & 1 & 0 \\ L_x & 0 & 1 \end{bmatrix} \xi^r. \end{aligned} \tag{88}$$

Compass Measurements

A compass will measure the direction of true north or, equivalently, the global x-axis with respect to the local frame. This axis is the first column of $R^\top$ and is a right-invariant measurement

$$z_m = R^\top \begin{bmatrix} 1 \\ 0 \end{bmatrix} + w_m, \qquad w_m \sim \mathcal{N}(0_2, \Sigma_m) \tag{89}$$

that can be put into right-invariant form as

$$\begin{bmatrix} z_m \\ 1 \end{bmatrix} = \begin{bmatrix} R^\top & -R^\top p \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} w_m \\ 0 \end{bmatrix} \triangleq X^{-1} b_m + \tilde{w}_m \tag{90}$$

and, thus, linearized as

$$\begin{aligned} H_m \xi^r = \Pi \xi^{r\wedge} b_m &= \Pi \begin{bmatrix} 0 & -\xi_\theta^r & \xi_x^r \\ \xi_\theta^r & 0 & \xi_y^r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \xi^r. \end{aligned} \tag{91}$$

Note that in some cases, a compass will point to true north with a slight offset, which can be modeled using a linear combination of the columns of $R^\top$. In this scenario, rather than use $b_m = [1 \;\; 0 \;\; 0]^\top$, a slight modification would be $b_m = [m_x \;\; m_y \;\; 0]^\top$ for some m_x and m_y.

Results

There is a performance difference between the right and left InEKFs that is largely dependent on the given type of measurement and measurement noise. To illustrate this, we run the above example on 0.5 s of simulated data, with measurements and controls sampled at 100 Hz.

Both the right and the left filters are implemented with only left measurements, in this case, GPS. These results are in the first four columns of Figure 5. The right and left filters are then implemented with only right measurements, that is, the compass and landmark sensors, with results shown in the last four columns of Figure 5. Two right measurements are used to ensure full observably. Each measurement has a $\Sigma = \mathrm{diag}(0.01, 0.01)$. Each scenario is executed 100 times, with random initial starting points sampled from $\mathcal{N}^\ell(I, \mathrm{diag}(25^\circ, 1, 1))$. The black dotted line in these plots is the ground truth value, except that the black dotted line in the Mahalanobis distance plots is the 99th percentile mark of a three-degrees-of-freedom chi-squared distribution.

While both filters converged within half a second or so, there are performance improvements when using a left filter with left measurements and a right filter with right measurements. Most notable is the large Mahalanobis distance when using a mixed filter and measurement combination. This implies that the mixed filter has an estimated covariance that is much too confident about the resulting estimate, even though the states converge nearly as well as a nonmixed filter. Overconfidence can have severe consequences if the estimated covariance is used in a planning or SLAM algorithm as a confidence interval for the state estimate.

While not shown, in the case where there are both right- and left-invariant measurements, the difference between the filters is much smaller and often insignificant. In this scenario, which filter is chosen will be dependent on many factors, including, but not limited to, which measurements observe the most important states, which measurements have the smallest covariance, and the process model differences in the "Process Model" section.

Imperfect InEKF Example

It is often the case that part of the state to be tracked does not fit into a Lie group structure. In these scenarios, an imperfect InEKF can be used to track the Euclidean states alongside the Lie group. This has often been used for tracking IMU biases. We note that in 3D, the current state of the art for invariant filters and bias tracking leverages the two-frame group [39], which we briefly discuss in the next section. However, in 2D, due to the angular velocity bias being a singleton and not fitting in the two-frame group, the imperfect InEKF still must be used to some extent, and we show its usage here. The approach described in [43] also outlines an approach that is similar to the imperfect InEKF. Another alternative is the equivariant filter [44], [45], [46], which also exploits a symmetry property but is not limited to Lie groups. It is shown in [45] that the equivariant filter is equivalent to the InEKF when the state and output manifold are Lie groups. It is shown in [47] that the equivariant filter can be used to effectively estimate biases.

We track orientation R, position p, and velocity v, modeled as an element of the Lie group $SE_2(2)$ or the group of double direct isometries [9]. An element $X \in SE_2(2)$ is of the form

$$X \triangleq \begin{bmatrix} R & v & p \\ 0_{1\times 2} & 1 & 0 \\ 0_{1\times 2} & 0 & 1 \end{bmatrix}, \quad \xi^{\wedge} = \begin{bmatrix} \xi_\theta \\ \xi_v \\ \xi_p \end{bmatrix}^{\wedge} = \begin{bmatrix} \xi_\theta^{\wedge} & \xi_v & \xi_p \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathrm{Ad}_X \xi = \begin{bmatrix} 1 & 0 & 0 \\ -v^{\star} & R & 0 \\ -p^{\star} & 0 & R \end{bmatrix} \xi, \quad \text{where } \xi_\theta^{\wedge} = \begin{bmatrix} 0 & -\xi_\theta \\ \xi_\theta & 0 \end{bmatrix}. \tag{92}$$

We model the IMU measurements with Gaussian noise and the IMU bias with Brownian motion, as

$$\omega = \tilde{\omega} + b_\omega + w_\omega, \qquad w_\omega \sim \mathcal{N}(0_{1\times 1}, \Sigma_\omega \delta(t-\tau)) \tag{93}$$

$$a = \tilde{a} + b_a + w_a, \qquad w_a \sim \mathcal{N}(0_{2\times 1}, \Sigma_a) \tag{94}$$

$$\dot{b}_\omega = w_{b\omega}, \qquad w_{b\omega} \sim \mathcal{N}(0_{1\times 1}, \Sigma_{b\omega} \delta(t-\tau)) \tag{95}$$

$$\dot{b}_a = w^{ba}, \qquad w_{ba} \sim \mathcal{N}(0_{2\times 1}, \Sigma_{ba} \delta(t-\tau)) \tag{96}$$

with $\omega, b_\omega \in \mathbb{R}$ the angular velocity and its bias, respectively, and $a, b_a \in \mathbb{R}^2$ the linear acceleration and its bias, respectively. Thus, in addition to the error η^r or η^ℓ, we track $\zeta \in \mathbb{R}^3$, defined as

$$\zeta = \begin{bmatrix} \zeta_\omega \\ \zeta_a \end{bmatrix} = \begin{bmatrix} \hat{b}_\omega - b_\omega \\ \hat{b}_a - b_a \end{bmatrix}. \tag{97}$$

The error state is therefore $[\xi^\top \ \zeta^\top]^\top$. Along with the augmented error, we need an augmented adjoint. Since the bias is updated via addition, which is commutative, we have

$$\mathrm{Ad}_{X,b} = \begin{bmatrix} \mathrm{Ad}_X & 0 \\ 0 & I \end{bmatrix}. \tag{98}$$

Finally, in the update step, the Kalman gain will be split as $K = [K^{\xi\top} \ K^{\zeta\top}]^\top$, where K^ξ will be used to update $\hat{X}$ using the matrix exponential and multiplication and K^ζ will update $[\hat{b}_\omega \ \hat{b}_a^\top]^\top$ using vector addition.

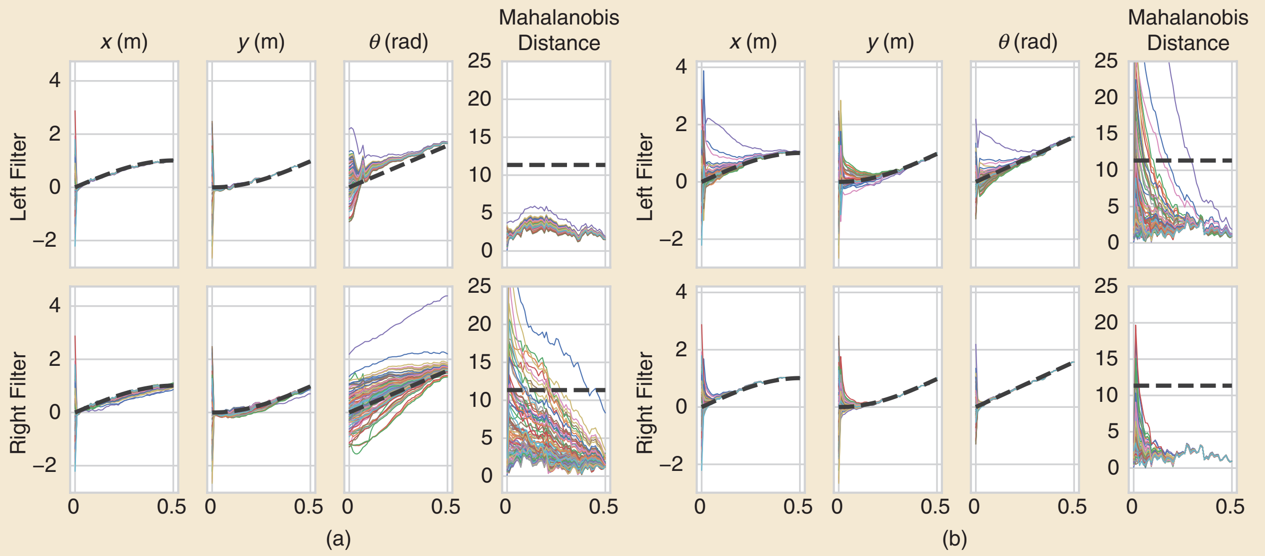

FIGURE 5 A comparison of right and left InEKFs with right and left measurements. Each light line represents a filter implementation, with random initial starting points sampled from $\mathcal{N}^l(I, \mathrm{diag}(25°, 1, 1))$, and the dashed black line represents the true state. In the Mahalanobis distance plot, the dashed black line represents the 99th percentile of the distribution. The filters are run for 0.5 s at 100 Hz. The first four columns are updated with only GPS data, a left-invariant measurement, and the last four with landmark and compass data, right-invariant measurements. Note the improved performance when the adjoint does not have to be used to convert measurements. The (a) left measurements and (b) right measurements.

Process Model

The continuous system dynamics are therefore [18]

$$\begin{aligned}\dot{R} &= R\,\tilde{w}^{\wedge} = R(\omega - w_{\omega} - b_{\omega})^{\wedge}\\ \dot{v} &= R\tilde{a} = R(a - w_a - b_a)\\ \dot{p} &= v\\ \dot{b}_{\omega} &= w_{b\omega}\\ \dot{b}_a &= w_{ba}.\end{aligned} \tag{99}$$

It is important to note that the process model has covariance Q along with joint noise vector w, as follows:

$$Q = \text{block\_diag}(\Sigma_{\omega}, \Sigma_a, 0_{3\times 3}, \Sigma_{b\omega}, \Sigma_{ba}) \tag{100}$$

$$w = [w_{\omega}^{\top}\ \ w_a^{\top}\ \ 0_{3\times 1}^{\top}\ \ w_{b\omega}^{\top}\ \ w_{ba}^{\top}]^{\top}. \tag{101}$$

Group Affine

The deterministic no-bias version of the dynamics

$$\begin{aligned}\frac{d}{dt}X &= \begin{bmatrix} R\omega^{\wedge} & Ra & v\\ 0_{1\times 2} & 0 & 0\\ 0_{1\times 2} & 0 & 0\end{bmatrix} - \begin{bmatrix} R & v & p\\ 0_{1\times 2} & 1 & 0\\ 0_{1\times 2} & 0 & 1\end{bmatrix}\begin{bmatrix} (w_{\omega})^{\wedge} & w_a & 0\\ 0_{1\times 2} & 0 & 0\\ 0_{1\times 2} & 0 & 0\end{bmatrix}\\ &\triangleq f_u(X) - X\tilde{w}^{\wedge}\end{aligned} \tag{102}$$

are group affine since

$$\begin{aligned}&f_u(X)Y + Xf_u(Y) - Xf_u(I)Y\\ &= \begin{bmatrix} R_x\tilde{\omega}^{\wedge}R_y & R_x\tilde{\omega}^{\wedge}v_y + R_x\tilde{a} & R_x\tilde{\omega}^{\wedge}p_y + v_x\\ 0_{1\times 3} & 0 & 0\\ 0_{1\times 3} & 0 & 0\end{bmatrix}\\ &+\begin{bmatrix} R_xR_y\,\tilde{\omega}^{\wedge} & R_xR_y\,\tilde{a} & R_xv_y\\ 0_{1\times 3} & 0 & 0\\ 0_{1\times 3} & 0 & 0\end{bmatrix}\\ &-\begin{bmatrix} R_x\tilde{\omega}^{\wedge}R_y & R_x(\tilde{\omega}^{\wedge}v_y + a) & R_x\tilde{\omega}^{\wedge}p_y\\ 0_{1\times 3} & 0 & 0\\ 0_{1\times 3} & 0 & 0\end{bmatrix}\\ &= \begin{bmatrix} R_xR_y\,\tilde{\omega}^{\wedge} & R_xR_y\,\tilde{a} & R_xv_y + v_x\\ 0_{1\times 3} & 0 & 0\\ 0_{1\times 3} & 0 & 0\end{bmatrix} = f_u(XY).\end{aligned} \tag{103}$$

Thus, the use of the InEKF on the Lie group portion of the state is justified.

Linearization

When using an imperfect InEKF, rather than use the formulas for the error dynamics g, the error dynamics must be linearized by hand. First, recall from (32) and (33) that η^r and η^{ℓ} are of the form

$$\eta^r \triangleq \begin{bmatrix} \eta_{\theta}^r & \eta_v^r & \eta_p^r\\ 0_{1\times 2} & 1 & 0\\ 0_{1\times 2} & 0 & 1\end{bmatrix} = \begin{bmatrix} \hat{R}R^{\top} & \hat{v} - \hat{R}R^{\top}v & \hat{p} - \hat{R}R^{\top}p\\ 0_{1\times 2} & 1 & 0\\ 0_{1\times 2} & 0 & 1\end{bmatrix} \tag{104}$$

$$\eta^{\ell} \triangleq \begin{bmatrix} \eta_{\theta}^{\ell} & \eta_v^{\ell} & \eta_p^{\ell}\\ 0_{1\times 2} & 1 & 0\\ 0_{1\times 2} & 0 & 1\end{bmatrix} = \begin{bmatrix} R^{\top}\hat{R} & R^{\top}(\hat{v} - v) & R^{\top}(\hat{p} - p)\\ 0_{1\times 2} & 1 & 0\\ 0_{1\times 2} & 0 & 1\end{bmatrix}. \tag{105}$$

For tutorial purposes, in the following, we show the detailed calculations used to compute the error state linear dynamics in the joint Lie algebra and Euclidean space, first for the right error, then the left error. Using the fact that on $SO(2)$, $a^{\wedge\top} = -a^{\wedge}$, and neglecting terms of order $O(|\xi_{\theta}^r||\zeta_{\omega}|)$ and using the expansion $\eta^r \approx I + \xi^{r\wedge}$, the right rotational error dynamics are

$$\begin{aligned}\frac{d}{dt}\eta_{\theta}^r &= \frac{d}{dt}\hat{R}R^{\top} = \left(\frac{d}{dt}\hat{R}\right)R^{\top} + \hat{R}\frac{d}{dt}R^{\top}\\ &= \hat{R}(\omega - \hat{b}_{\omega})^{\wedge}R^{\top} - \hat{R}(\omega - b_{\omega})^{\wedge}R^{\top}\\ &= -\hat{R}(\hat{b}_{\omega} - b_{\omega})^{\wedge}R^{\top}\\ &= -\hat{R}R^{\top}\zeta_{\omega}^{\wedge}\\ &= -\eta_{\theta}^r\zeta_{\omega}^{\wedge} \approx -(I + \xi_{\theta}^r)\zeta_{\omega}^{\wedge}\\ &\approx -\zeta_{\omega}^{\wedge}.\end{aligned} \tag{106}$$

Using the identity

$$a^{\wedge}b = \begin{bmatrix} 0 & -a\\ a & 0\end{bmatrix}\begin{bmatrix} b_1\\ b_2\end{bmatrix} = \begin{bmatrix} -b_2\\ b_1\end{bmatrix}a = b^{\star}a \tag{107}$$

where $a \in \mathbb{R}$ and $b = [b_1\ \ b_2]^{\top} \in \mathbb{R}^2$, gives

$$\begin{aligned}\frac{d}{dt}\eta_v^r &= \frac{d}{dt}(\hat{v} - \hat{R}R^{\top}v)\\ &= \frac{d}{dt}\hat{v} - \frac{d}{dt}(\hat{R}R^{\top})v - \hat{R}R^{\top}\frac{d}{dt}v\\ &= \hat{R}(a - \hat{b}_a) + \zeta_{\omega}^{\wedge}v - \hat{R}R^{\top}R(a - b_a)\\ &\approx -\hat{R}\zeta_a + \hat{v}^{\star}\zeta_{\omega}.\end{aligned} \tag{108}$$

The right positional error is derived using $\eta \approx I + \xi^{\wedge}$, as

$$\begin{aligned}\frac{d}{dt}\eta_p^r &= \frac{d}{dt}(\hat{p} - \hat{R}R^{\top}p)\\ &= \frac{d}{dt}\hat{p} - \frac{d}{dt}(\hat{R}R^{\top})p - \hat{R}R^{\top}\frac{d}{dt}p\\ &= \hat{v} + \zeta_{\omega}^{\wedge}p - \hat{R}R^{\top}v\\ &= \eta_v^r + p^{\star}\zeta_{\omega} \approx \xi_v^r + \hat{p}^{\star}\zeta_{\omega}.\end{aligned} \tag{109}$$

Since the deterministic dynamics of ζ are zeros, we have

$$\frac{d}{dt}\begin{bmatrix}\xi^r\\ \zeta\end{bmatrix} = A^r\begin{bmatrix}\xi^r\\ \zeta\end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0\\ 0 & 0 & 0 & \hat{v}^{\star} & -\hat{R}\\ 0 & I & 0 & \hat{p}^{\star} & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\xi_{\theta}^r\\ \xi_v^r\\ \xi_p^r\\ \zeta_{\omega}\\ \zeta_a\end{bmatrix}. \tag{110}$$

Following a similar process for η^{ℓ}, noting that matrices commute in $SO(2)$ and neglecting terms of order $O(|\xi_{\theta}^{\ell}||\zeta_{\omega}|)$, gives

$$\begin{aligned}\frac{d}{dt}\eta_{\theta}^{\ell} &= \frac{d}{dt}R^{\top}\hat{R} = \frac{d}{dt}(R^{\top})\hat{R} + R^{\top}\frac{d}{dt}\hat{R}\\ &= -(\omega - b_{\omega})^{\wedge}R^{\top}\hat{R} + R^{\top}\hat{R}(\omega - \hat{b}_{\omega})^{\wedge}\\ &= R^{\top}\hat{R}(b_{\omega} - \hat{b}_{\omega})^{\wedge}\\ &= -\eta_{\theta}^{\ell}\zeta_{\omega}^{\wedge} \approx -(I + \xi_{\theta}^{\ell})\zeta_{\omega}^{\wedge}\\ &= -\zeta_{\omega}^{\wedge}.\end{aligned} \tag{111}$$

The left velocity error dynamics are derived as

$$\begin{aligned}\frac{d}{dt}\eta_v^{\ell} &= \frac{d}{dt}R^{\top}(\hat{v} - v)\\ &= \frac{d}{dt}(R^{\top})(\hat{v} - v) + R^{\top}\frac{d}{dt}(\hat{v} - v)\\ &= -(\omega - b_{\omega})^{\wedge}R^{\top}(\hat{v} - v) + R^{\top}(\hat{R}(a - \hat{b}_a) - R(a - b_a))\\ &= -(\omega - b_{\omega})^{\wedge}\eta_v^{\ell} + \eta_{\theta}^{\ell}(a - \hat{b}_a) - a + b_a\\ &\approx -(\omega - b_{\omega})^{\wedge}\xi_v^{\ell} + (I + \xi_{\theta}^{\ell\wedge})(a - \hat{b}_a) - a + b_a\\ &= -(\omega - b_{\omega})^{\wedge}\xi_v^{\ell} + \xi_{\theta}^{\ell\wedge}(a - \hat{b}_a) + a - \hat{b}_a - a + b_a\\ &= -(\omega - b_{\omega})^{\wedge}\xi_v^{\ell} + (a - \hat{b}_a)^{\star}\xi_{\theta}^{\ell} - \zeta_a.\end{aligned} \tag{112}$$

The left positional error dynamics are

$$\begin{aligned}\frac{d}{dt}\eta_p^\ell &= \frac{d}{dt}R^\top(\hat{p}-p)\\ &= \frac{d}{dt}(R^\top)(\hat{p}-p)+R^\top\frac{d}{dt}(\hat{p}-p)\\ &= -(\omega-b_\omega)^\wedge R^\top(\hat{p}-p)+R^\top(\hat{v}-v)\\ &= -(\omega-b_\omega)^\wedge\eta_p^\ell+\eta_v^\ell\\ &\approx -(\omega-b_\omega)^\wedge\xi_p^\ell+\xi_v^\ell.\end{aligned} \tag{113}$$

The resulting continuous linear system therefore has the state update matrix

$$A^\ell = \begin{bmatrix} 0 & 0 & 0 & -1 & 0\\ (a-\hat{b}_a)^\star & -(\omega-\hat{b}_\omega)^\wedge & 0 & 0 & -I\\ 0 & I & -(\omega-\hat{b}_\omega)^\wedge & 0 & 0\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{bmatrix}. \tag{114}$$

GPS Measurements

Similar to the odometry example, a GPS measurement is a left-invariant measurement, with a near-identical linearized H that includes additional zeros for the velocity and bias states:

$$H_p\begin{bmatrix}\xi^\ell\\ \zeta\end{bmatrix} \triangleq [0_{2\times 3}\ \ -I\ \ 0_{2\times 3}]\begin{bmatrix}\xi^\ell\\ \zeta\end{bmatrix}. \tag{115}$$

Compass Measurements

Similarly, the compass measurement is a right-invariant measurement with additional zeros to obtain

$$H_m\begin{bmatrix}\xi^r\\ \zeta\end{bmatrix} \triangleq \begin{bmatrix}0 & 0_{1\times 7}\\ 1 & 0_{1\times 7}\end{bmatrix}\begin{bmatrix}\xi^r\\ \zeta\end{bmatrix}. \tag{116}$$

Velocity Measurements

Velocity can be measured in a number of ways, from pressure sensors on aircraft to Doppler velocity logs on underwater vehicles. Generally, velocity is measured in the local frame, resulting in a right-invariant measurement:

$$z_v = R^\top v + w_v, \qquad w_v \sim N(0_{2\times 1}, \Sigma_v) \tag{117}$$

which can be rewritten in right-invariant form as

$$\begin{bmatrix}z_v\\ 1\end{bmatrix} = \begin{bmatrix}R^\top & -R^\top v & R^\top p\\ 0_{1\times 3} & 1 & 0\\ 0_{1\times 3} & 0 & 1\end{bmatrix}\begin{bmatrix}0_{3\times 1}\\ -1\\ 0\end{bmatrix} + \begin{bmatrix}w_v\\ 0_{2\times 1}\end{bmatrix} \triangleq X^{-1}b_v + \tilde{w}_v. \tag{118}$$

Therefore, from (65),

$$\begin{aligned}\Pi\xi^{r\wedge}b_v = \Pi\begin{bmatrix}\xi_\theta^\wedge & \xi_v & \xi_p\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}0_{3\times 1}\\ -1\\ 0\end{bmatrix} &= [0_{2\times 1}\ \ -I\ \ 0_{2\times 2}]\xi^r\\ \Rightarrow H_v\begin{bmatrix}\xi^r\\ \zeta\end{bmatrix} &\triangleq [0_{2\times 1}\ \ -I\ \ 0_{2\times 5}]\begin{bmatrix}\xi^r\\ \zeta\end{bmatrix}.\end{aligned} \tag{119}$$

Results

To illustrate the effectiveness of the imperfect InEKF, we instantiate both the left and right versions on 0.5 s of simulation data with the above process and measurement models, with measurements and controls sampled at 100 Hz. The results can be seen in Figure 6. Note that the biases do not converge in this simulation since the system is not fully observable [18].

Due to the simplicity of this example built on $SE_2(2)$, we find that generally the quaternion extended Kalman filter

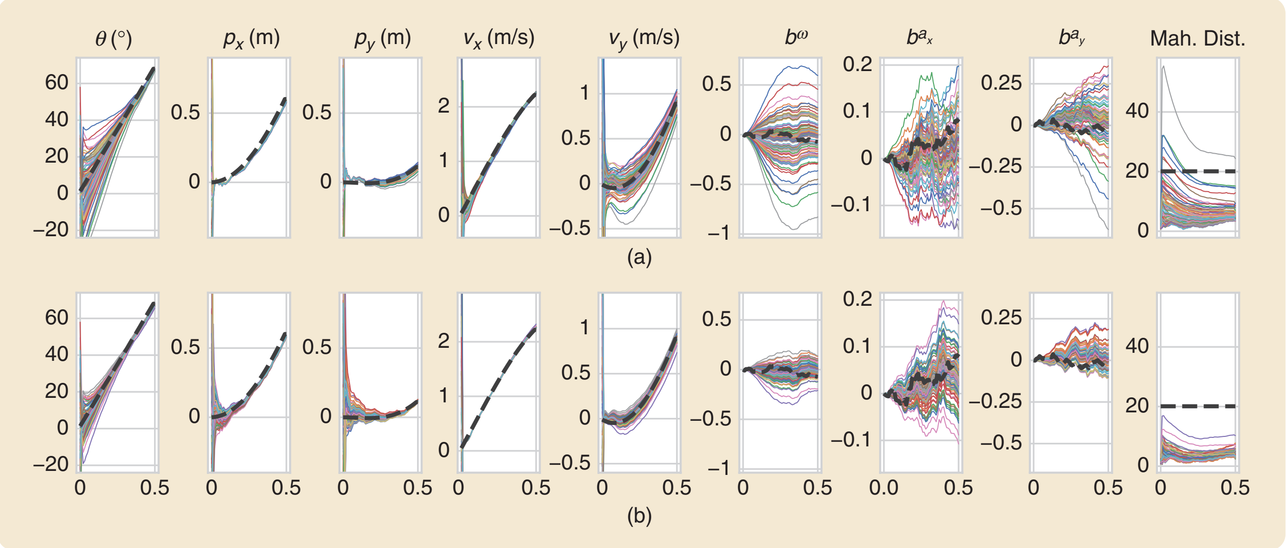

FIGURE 6 A comparison of right and left imperfect InEKFs with GPS, magnetometer, and velocity sensor measurements. Each light line represents a filter implementation, with random initial starting points sampled from $\mathcal{N}^l(I, \mathrm{diag}(10°, 4, 4, 1, 1))$, and the dashed black line represents the true state. In the Mahalanobis distance plot, the dashed black line represents the 99th percentile of the distribution. The filters are run for 0.5 s at 100 Hz. Note that the biases do not converge in this simulation since the system is not fully observable [18]. The (a) left filter and (b) right filter.

(QEKF) and other ESKFs perform similarly to the imperfect InEKF. This can be seen as the lower limit for the imperfect InEKF; in our experience, it performs at least as well as other ESKFs, but in most scenarios, it has improved performance [18], [20].

Furthermore, notice the improved performance of states measured with a right-invariant measurement when used in a right filter. For example, velocity and heading, measured with a right-invariant measurement, converge faster with a right filter, while position, measured with a left-invariant measurement, converges faster with a left-invariant filter. If there are specific measured states that are more mission critical, the filter type should be chosen to match those states' measurement types. In this example, if position were extremely critical, a left filter should likely be chosen.

Two-Frame InEKF

When using $SE(2)$ or $SE(3)$, the position p_{gl}^{g} is in the global or "fixed" frame. In some cases, there are also vectors that need to be tracked in the local or "body" frame, such as IMU biases. In these scenarios, the two-frame group can be used to track both in a Lie group.

The two-frame group tracks a tuple (R, x, X), with $R \in SO(d)$, $x \in \mathbb{R}^{d \times M}$, and $X \in \mathbb{R}^{d \times N}$, where vectors in x are represented in the global frame and vectors in X in the local frame. The identity is given by $(I_d, 0, 0)$, the group action is defined as

$$\begin{bmatrix} R_i \\ x_i \\ X_i \end{bmatrix} \cdot \begin{bmatrix} R_j \\ x_j \\ X_j \end{bmatrix} = \begin{bmatrix} R_i R_j \\ x_i + R_i x_j \\ X_j + R_j^{-1} x_i \end{bmatrix} \tag{120}$$

and the inverse is defined as

$$\begin{bmatrix} R \\ x \\ X \end{bmatrix}^{-1} = \begin{bmatrix} R^{-1} \\ -R^{-1} x \\ -RX \end{bmatrix}. \tag{121}$$

From these properties, it can be shown that the two-frame group is a group and can be used in the InEKF. Its usage is beyond the tutorial nature of this article, but we point the readers to other literature for more information [39].

CONCLUSION

The recently developed InEKF has been experimentally shown to be superior to the standard EKF and ESKF. However, there does not appear to be an accessible explanation of the difference between the right and left InEKFs and the tradeoffs involved in choosing one of those implementations. We hope that this tutorial filled that gap. We have also included a full derivation of the associated error state dynamics, with the hope that these derivations will help others implement the InEKF for their specific applications. We also pointed the readers to other literature for more state-of-the-art developments in invariant Kalman filtering, such as the two-frame group [39], IMU preintegration [23], [24], reachability [22], and observability [21]. Finally, we have developed an open source InEKF library with both Python and C++ interfaces for quick implementation of both the right and left InEKFs. Future work includes expansion of this library, along with more experimental results comparing the right and left InEKF filters with each other and with equivalent ESKF implementations.

AUTHOR INFORMATION

Easton R. Potokar (epotokar@andrew.cmu.edu) received the B.S. and M.S. degrees in applied mathematics and electrical engineering, respectively, from Brigham Young University, where he worked in the Field Robotic Systems Lab. There, he helped develop HoloOcean, an open source marine robotics simulator, and he worked on applications of the invariant extended Kalman filter to underwater localization. He is currently pursuing the Ph.D. degree in robotics from Carnegie Mellon University, Pittsburgh, PA 15213 USA, in the Robot Perception Lab, where he works on GPS-denied localization.

Randal W. Beard received the B.S. degree in electrical engineering from the University of Utah, Salt Lake City, in 1991, and he received the M.S. degree in electrical engineering in 1993, the M.S. degree in mathematics in 1994, and the Ph.D. degree in electrical engineering in 1995, all from Rensselaer Polytechnic Institute, Troy, NY, USA. Since 1996, he has been with the Department of Electrical and Computer Engineering, Brigham Young University (BYU), Provo, UT 84602 USA, where he is currently a professor. His primary research focus is autonomous systems, unmanned air vehicles, and multiple-vehicle coordination and control. He is a former associate editor for *IEEE Transactions on Automatic Control*, *Journal of Intelligent and Robotics Systems*, and *IEEE Control Systems*. He currently holds the Steven V. White University Professorship at BYU. He is a Fellow of IEEE and the American Institute of Aeronautics and Astronautics.

Joshua G. Mangelson received the B.S. degree in electrical engineering from Brigham Young University in 2014 and the M.S. and Ph.D. degrees in robotics from the University of Michigan in 2016 and 2019, respectively. After completing his doctorate, he served as a postdoctoral fellow at Carnegie Mellon University before joining the Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602 USA, in 2020. His qualifications include demonstrated expertise in robotic perception, mapping, and localization, with a particular focus on marine robotics. He has conducted field trials in various locations around the world, including Hawaii, USA; San Diego, CA, USA; Boston, MA, USA; northern Michigan, USA; and Utah. In 2018, his work on multirobot mapping received the Best Multi-Robot Paper Award at the IEEE International Conference on Robotics and Automation and first place in the IEEE OCEANS Student Poster Competition. He is currently an associate editor for *International Journal of Robotics Research*. He received the Office of Naval Research Young Investigator Award in 2024. He is a Member of IEEE.

REFERENCES

[1] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Trans. ASME—J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960, doi: 10.1115/1.3662552.

[2] J. Deyst and C. Price, “Conditions for asymptotic stability of the discrete minimum-variance linear estimator,” *IEEE Trans. Autom. Control*, vol. 13, no. 6, pp. 702–705, Dec. 1968, doi: 10.1109/TAC.1968.1099024.

[3] S. Särkkä, *Bayesian Filtering and Smoothing* (Institute of Mathematical Statistics Textbooks). Cambridge, U.K.: Cambridge Univ. Press, 2013.

[4] S. Roumeliotis, G. Sukhatme, and G. Bekey, “Circumventing dynamic modeling: Evaluation of the error-state Kalman filter applied to mobile robot localization,” in *Proc. IEEE Int. Conf. Robot. Automat. (Cat. No.99CH36288C)*, vol. 2, May 1999, pp. 1656–1663, doi: 10.1109/ROBOT.1999.772597.

[5] S. Bonnabel, “Left-invariant extended Kalman filter and attitude estimation,” in *Proc. IEEE Conf. Decision Control*, New Orleans, LA, USA, Dec. 2007, pp. 1027–1032.

[6] S. Bonnabel, P. Martin, and E. Salaün, “Invariant extended Kalman filter: Theory and application to a velocity-aided attitude estimation problem,” in *Proc. 48h IEEE Conf. Decision Control (CDC)*, Dec. 2009, pp. 1297–1304.

[7] A. Barrau and S. Bonnabel, “The invariant extended Kalman filter as a stable observer,” *IEEE Trans. Automat. Control*, vol. 62, no. 4, pp. 1797–1812, Apr. 2017, doi: 10.1109/TAC.2016.2594085.

[8] A. Barrau and S. Bonnabel, “Invariant Kalman filtering,” *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, no. 1, pp. 237–257, 2018, doi: 10.1146/annurev-control-060117-105010.

[9] A. Barrau, “Non-linear state error based extended Kalman filters with applications to navigation,” Ph.D. dissertation, Mines Paristech, Paris, France, 2015.

[10] M. Brossard, S. Bonnabel, and A. Barrau, “Invariant Kalman filtering for visual inertial SLAM,” in *Proc. Int. Conf. Inf. Fusion (FUSION)*, Cambridge, U.K., Jul. 2018, pp. 2021–2028.

[11] Y. Song, Z. Zhang, J. Wu, Y. Wang, L. Zhao, and S. Huang, “A right invariant extended Kalman filter for object based SLAM,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1316–1323, Apr. 2022, doi: 10.1109/LRA.2021.3139370.

[12] K. Wu, T. Zhang, D. Su, S. Huang, and G. Dissanayake, “An invariant-EKF VINS algorithm for improving consistency,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vancouver, Canada, Sep. 2017, pp. 1578–1585.

[13] Y. Yang, C. Chen, W. Lee, and G. Huang, “Decoupled right invariant error states for consistent visual-inertial navigation,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1627–1634, Apr. 2022, doi: 10.1109/LRA.2021.3140054.

[14] X. Li, H. Jiang, X. Chen, H. Kong, and J. Wu, “Closed-form error propagation on SE_n (3) group for invariant EKF with applications to VINS,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10705–10712, Oct. 2022, doi: 10.1109/LRA.2022.3194684.

[15] M. Barczyk and A. F. Lynch, “Invariant observer design for a helicopter UAV aided inertial navigation system,” *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 791–806, May 2013, doi: 10.1109/TCST.2012.2195495.

[16] M. Barczyk and A. F. Lynch, “Invariant extended Kalman filter design for a magnetometer-plus-GPS aided inertial navigation system,” in *Proc. IEEE Conf. Decision Control*, Orlando, FL, USA, Dec. 2011, pp. 5389–5394.

[17] N. Y. Ko, W. Youn, I. H. Choi, G. Song, and T. S. Kim, “Features of invariant extended Kalman filter applied to unmanned aerial vehicle navigation,” *Sensors*, vol. 18, no. 9, pp. 2855, Sep. 2018, doi: 10.3390/s18092855.

[18] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, “Contact-aided invariant extended Kalman filtering for robot state estimation,” *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 402–430, 2020, doi: 10.1177/0278364919894385.

[19] S. Teng, M. W. Mueller, and K. Sreenath, “Legged robot state estimation in slippery environments using invariant extended Kalman filter with velocity update,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2021, pp. 3104–3110, doi: 10.1109/ICRA48506.2021.9561313.

[20] E. Potokar, K. Norman, and J. G. Mangelson, “Invariant extended Kalman filtering for underwater navigation,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5792–5799, Jul. 2021, doi: 10.1109/LRA.2021.3085167.

[21] A. Barrau and S. Bonnabel, “An EKF-SLAM algorithm with consistency properties,” 2016, *arXiv:1510.06263*.

[22] A. Barrau and S. Bonnabel, “Extended Kalman filtering with nonlinear equality constraints: A geometric approach,” *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2325–2338, Jun. 2020.

[23] A. Barrau and S. Bonnabel, “Linear observed systems on groups,” *Syst. Control Lett.*, vol. 129, pp. 36–42, Jul. 2019.

[24] M. Brossard, A. Barrau, P. Chauchat, and S. Bonnabel, “Associating uncertainty to extended poses for on lie group IMU preintegration with rotating earth,” *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 998–1015, Apr. 2022, doi: 10.1109/TRO.2021.3100156.

[25] K. Wu, T. Zhang, D. Su, S. Huang, and G. Dissanayake, “An invariant-EKF VINS algorithm for improving consistency,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2017, pp. 1578–1585, doi: 10.1109/IROS.2017.8205965.

[26] A. Barrau and S. Bonnabel, “Three examples of the stability properties of the invariant extended Kalman filter *,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 431–437, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.061.

[27] R. Hartley, “Contact-aided state estimation on Lie groups for legged robot mapping and control,” thesis, Univ. Michigan, Ann Arbor, MI, USA, 2019.

[28] J. Arsenault, “Practical considerations and extensions of the invariant extended Kalman filtering framework,” thesis, McGill Univ., Montreal, QC, Canada, 2020.

[29] M. P. d Carmo, *Riemannian Geometry*. Basel, Switzerland: Birkhäuser, 1993.

[30] T. D. Barfoot and P. T. Furgale, “Associating uncertainty with three-dimensional poses for use in estimation problems,” *IEEE Trans. Robot.*, vol. 30, no. 3, pp. 679–693, Jun. 2014, doi: 10.1109/TRO.2014.2298059.

[31] B. C. Hall, *Lie Groups, Lie Algebras, and Representations an Elementary Introduction*. Cham, Switzerland: Springer, 2015.

[32] J. Solà, J. Deray, and D. Atchuthan, “A micro Lie theory for state estimation in robotics,” Dec. 2021, *arXiv:1812.01537*.

[33] E. Eade, “Lie groups for 2D and 3D transformations,” 2022. Accessed: Oct. 1, 2024. [Online]. Available: https://www.ethaneade.com/lie.pdf

[34] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen, “Stochastic stability of the discrete-time extended Kalman filter,” *IEEE Trans. Autom. Control*, vol. 44, no. 4, pp. 714–728, Apr. 1999, doi: 10.1109/9.754809.

[35] B. F. La Scala, R. R. Bitmead, and M. R. James, “Conditions for stability of the extended Kalman filter and their application to the frequency tracking problem,” *Math. Control, Signals Syst.*, vol. 8, no. 1, pp. 1–26, Mar. 1995, doi: 10.1007/BF01212364.

[36] B. Bell and F. Cathey, “The iterated Kalman filter update as a Gauss-Newton method,” *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 294–297, Feb. 1993, doi: 10.1109/9.250476.

[37] J. Solà, “Quaternion kinematics for the error-state KF,” 2015, *arXiv:1711.02508*.

[38] R. V. Vitali, R. S. McGinnis, and N. C. Perkins, “Robust error-state Kalman filter for estimating IMU orientation,” *IEEE Sensors J.*, vol. 21, no. 3, pp. 3561–3569, Feb. 2021, doi: 10.1109/JSEN.2020.3026895.

[39] A. Barrau and S. Bonnabel, “The geometry of navigation problems,” *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 689–704, Feb. 2023, doi: 10.1109/TAC.2022.3144328.

[40] J. G. Mangelson, M. Ghaffari, R. Vasudevan, and R. M. Eustice, “Characterizing the uncertainty of jointly distributed poses in the Lie algebra,” *IEEE Trans. Robot.*, vol. 36, no. 5, pp. 1371–1388, Oct. 2020, doi: 10.1109/TRO.2020.2994457.

[41] N. Leonard and P. Krishnaprasad, “Motion control of drift-free, left-invariant systems on Lie groups,” *IEEE Trans. Autom. Control*, vol. 40, no. 9, pp. 1539–1554, Sep. 1995, doi: 10.1109/9.412625.

[42] P. S. Maybeck, *Stochastic Models, Estimation and Control* (Mathematics in Science and Engineering). New York, NY, USA: Academic, 1979.

[43] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu, “Discrete extended Kalman filter on lie groups,” in *Proc. 21st Eur. Signal Process. Conf. (EUSIPCO)*, 2013, pp. 1–5.

[44] P. v Goor and R. Mahony, “An equivariant filter for visual inertial odometry,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2021, pp.14432–14438, doi: 10.1109/ICRA48506.2021.9561769.

[45] P. van Goor, T. Hamel, and R. Mahony, “Equivariant filter (EqF),” *IEEE Trans. Autom. Control*, vol. 68, no. 6, pp. 3501–3512, Jun. 2023, doi: 10.1109/TAC.2022.3194094.

[46] P. van Goor and R. Mahony, “EqVIO: An equivariant filter for visual-inertial odometry,” *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3567–3585, Oct. 2023, doi: 10.1109/TRO.2023.3289587.

[47] A. Fornasier, Y. Ng, C. Brommer, C. Böhm, R. Mahony, and S. Weiss, “Overcoming bias: Equivariant filter design for biased attitude estimation with online calibration,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 12118–12125, Oct. 2022, doi: 10.1109/LRA.2022.3210867.