Invariant Smoothing for Localization: Including the IMU Biases

Paul Chauchat, Silvère Bonnabel, Axel Barrau

Invariant Smoothing for Localization: Including the IMU Biases

Paul Chauchat[1], Silvère Bonnabel[2] and Axel Barrau[2,3]

***Abstract*—In this article we investigate smoothing (i.e., optimisation-based) estimation techniques for robot localization using an IMU aided by other localization sensors. We more particularly focus on Invariant Smoothing (IS), a variant based on the use of nontrivial Lie groups from robotics. We study the recently introduced Two Frames Group (TFG), and prove it can fit into the framework of Invariant Smoothing in order to better take into account the IMU biases, as compared to the state-of-the-art in robotics. Experiments based on the KITTI dataset show the proposed framework compares favorably to the state-of-the-art smoothing methods in terms of robustness in some challenging situations.**

I. INTRODUCTION

Lie group embeddings have become standard tools in navigation and mobile robotics over the last decade, see e.g., [1,18,40]. The use of $SE(3)$ and $SE(2)$ was pioneered by several authors in robotics, notably [2,34,38,45] and shown to gracefully fit unicycle-like models for wheeled robots moving in 2D. Extended Kalman filters based on those classical groups have been introduced and successfully used, see [10,30]Novel Lie groups have been introduced and proved to come with similar properties when using more complex motion models based on the (3D) IMU equations, namely $SE_2(3)$ introduced in [3,5], as well as the general group $SE_K(d)$ which was shown to resolve the consistency issues of the Extended Kalman filter in SLAM, see [3,4].

Those novel Lie groups came with a new class of estimators, under the name of Invariant Extended Kalman Filter (IEKF) [5,6]. The theoretical properties include convergence guarantees [5], consistency properties for SLAM [4,11], and have led to applications in various fields, notably robotics [6,13,27]–[29,36,39,42,46] and in the industry [6].

The success of the IEKF for models based on the IMU relies on the introduction of the Lie group of double spatial direct isometries $SE_2(3)$, or extended poses, in [3,5]. A thorough description of this group and its ability to capture uncertainty can be found in the recent paper [12]. However, IMU biases could not be properly included into this Lie group structure, and were treated as additional parameters leading to an "imperfect" invariant framework [3,42]. This limitation was greatly relieved by the introduction of the first systematic way of designing systems which fit the invariant framework, relying on the Two Frame Group (TFG) structure [7]. It encompasses the previously published groups, including $SE_2(3)$ and allows including the accelerometer bias.

Smoothing is now one of the most popular state estimation methods in robotics for simultaneous localisation and mapping (SLAM) and visual odometry, thanks to its reducing the consequences of linearisation errors [22,24,33]. This success broadened its application domains, and it is now often used in inertial navigation [31,32,43,48]. Leveraging the framework of Invariant filtering for smoothing, a new estimation algorithm was recently proposed, namely Invariant Smoothing (IS) [16], see also [43]. Similar to the IEKF [9,14], IS delivers "physically consistent" estimates [17].

This paper provides all the tools required to use the TFG structure in the smoothing framework, in order to tackle biased localization using biased IMUs. To evaluate the proposed framework, experiments were conducted based on data from the KITTI dataset [26], focusing in particular on the difficult problem of "robot kidnapping" [41], that is, localization with no prior information, known as "in-flight alignment" problem in inertial navigation when the sensors used are the IMU and GNSS [20,37,47]. Results show that, for this problem, using the KITTI dataset, IS based on the TFG favorably compares to state-of-the-art smoothing schemes [21,24], and to the former "imperfect" IS based on $SE_2(3)$, by proving more robust.

The main principles of Invariant Filtering and Smoothing are recalled in Section II. The considered application to localization is presented in Section III, as well as the presentation of the TFG and its properties for the considered problem. The proposed invariant smoother is detailed in Section IV, and the differences to the existing state-of-the-art smoothers are explained. Experimental results and comparisons are presented in Section V.

II. SMOOTHING ON LIE GROUPS

We first briefly recall the invariant filtering framework [5, 6]. The reader is referred to [1,40] for a general presentation. We consider a state $\chi \in G$, with G a Lie group of dimension q. Its Lie algebra $\mathfrak{g}$ is identified with $\mathbb{R}^q$. Thus we consider its exponential map to be defined as $\exp : \mathbb{R}^q \to G$. We denote its local inverse by log. We recall the notion of adjoint operator matrix of $\chi \in G$, $\mathbf{Ad}_\chi$, which satisfies

$$\forall \chi \in G, \xi \in \mathbb{R}^q,\ \chi^{-1} \exp(\xi) \chi = \exp(\mathbf{Ad}_\chi \xi) \tag{1}$$

Automorphisms are bijective maps $\phi : G \to G$ satisfying

$$\phi(\chi\eta) = \phi(\chi)\phi(\eta) \quad \text{for } \chi, \eta \in G. \tag{2}$$

[1]LIS-Lab, Aix-Marseille Université, Marseille, France paul.chauchat@lis-lab.fr

[2]Centre for Robotics, MINES Paris, PSL Research University, 60 Boulevard Saint-Michel, 75006 Paris, France silvere.bonnabel@mines-paristech.fr

[3]OFFROAD, 5 rue Charles de Gaulle, Alfortville, France axel@offroad.works

The Lie group Lie algebra correspondance, see [8], ensures for *any* automorphism ϕ there is $\mathbf{M} \in \mathbb{R}^{q\times q}$ so that

$$\forall(\chi,\xi) \in G\times\mathbb{R}^q,\ \phi(\chi\exp(\xi)) = \phi(\chi)\exp(\mathbf{M}\xi), \tag{3}$$

which is closely related to the *log-linearity* property of [5]. The operator $v \mapsto \chi^{-1}v\chi$ is easily checked to be a group automorphism, and we see indeed from (1) that $\mathbf{M} = \mathbf{Ad}_\chi$. We define random variables on Lie groups through the exponential, following [1,6,10,12,18]. The probability distribution $\chi \sim \mathscr{N}_L(\bar{\chi},\mathbf{P})$ for the random variable $\chi \in G$ is defined as

$$\chi = \bar{\chi}\exp(\xi),\ \xi \sim \mathscr{N}(\mathbf{0},\mathbf{P}), \tag{4}$$

In the following, we consider a discrete-time trajectory denoted as $(\chi_i)_i$ of the following system

$$\chi_0 \sim \mathscr{N}_L(\bar{\chi},\mathbf{P}_0),\quad \chi_{i+1} = f_i(\chi_i)\exp(w_i),\quad y_k = h_k(\chi_{I_k}) + \mathbf{n}_k \tag{5}$$

where f_i is the dynamics function, $\mathbf{P}_0 \in \mathbb{R}^{q\times q}$ the initial state error covariance, $\mathbf{w}_i, \mathbf{n}_k$ are white noises of covariance $\mathbf{Q}_i$ and $\mathbf{N}_k$ the observation noise covariance, and χ_{I_k} denotes a subset of the states which are involved in the measurements at t_k.

A. Smoothing on Lie groups

We first briefly recall the Invariant Smoothing (IS) framework introduced in [16]. Departing from a system of the form (5), the goal of smoothing is to find

$$(\chi_i)_i^* = \underset{(\chi_i)_{1\le i\le n}}{\operatorname{argmax}}\, \mathbb{P}((\chi_i)_i | y_0,\dots,y_n) \tag{6}$$

i.e., the maximum a posteriori (MAP) estimate of the trajectory. It is usually found through Gauss-Newton or Levenberg-Marquardt algorithms. First we devise a cost function associated to Problem (6) as the negative log likelihood

$$\mathscr{C} = -\log\big(\mathbb{P}((\chi_i)_{1\le i\le n} | y_0,\dots,y_n)\big)$$

that we seek to minimize. Given a current guess of the trajectory's states, $(\hat{\chi}_i)_i$, the cost function $\mathscr{C}$ is linearised and then the resulting linear problem is solved exactly, yielding a novel estimate, and so on until convergence. Since χ_i belongs to a Lie group, linearisation in IS is carried out as

$$\forall 1 \le i \le n \quad \chi_i = \hat{\chi}_i\exp(\xi_i). \tag{7}$$

where $(\xi_i)_i$ are the searched parameters that minimize the linearized cost. IS linearises the cost $\mathscr{C}$ as [16]

$$\begin{aligned}\tilde{\mathscr{C}} =& \|\mathbf{p}_0 + \xi_0\|^2_{\widetilde{\mathbf{P}}_0} \\ &+ \sum_i \|\hat{\mathbf{u}}_i - \mathbf{F}_i\xi_i + \xi_{i+1}\|^2_{\mathbf{Q}_i} + \sum_k \|\hat{\mathbf{n}}_k + \mathbf{H}_k\Xi\|^2_{\mathbf{N}_k}\end{aligned} \tag{8}$$

where we used the notation $\|\mathbf{Z}\|^2_{\mathbf{P}} = \mathbf{Z}^\top\mathbf{P}^{-1}\mathbf{Z}$, and where Ξ is the concatenation of $(\xi_i)_i$. (8) relies on the Baker-Campbell-Haussdorff formula [1] $\log(\exp(a)\exp(b)) = BCH(a,b)$. $\widetilde{\mathbf{P}}_0 = \mathbf{J}_0^{-1}\mathbf{P}_0\mathbf{J}_0^{-T}$, where $\mathbf{J}_0$ is the left Jacobian of the Lie group G [1,18], satisfying $BCH(\mathbf{p}_0,\xi) = \mathbf{p}_0 + \mathbf{J}_0\xi + o(\|\xi\|^2)$, $\mathbf{p}_0 = \log(\bar{\chi}_0^{-1}\hat{\chi}_0)$ with a prior $\bar{\chi}_0$, $\hat{\mathbf{u}}_i = \log(f_i(\hat{\chi}_i)^{-1}\hat{\chi}_{i+1})$, $\hat{\mathbf{n}}_k = \mathbf{y}_k - h_k(\hat{\chi}_{I_k})$, and $\mathbf{F}_i, \mathbf{H}_k$ are the (Lie group) Jacobians of f_i and h_k respectively. $\mathbf{H}_k$ was padded with zero blocks for the indices not contained in I_k. The principle of smoothing algorithms is to solve the linearized problem (8) in closed form, and to update the trajectory substituting the optimal ξ_i in (7). The problem is then relinearised at this new estimate until convergence.

B. Group-affine Dynamics and Invariant Smoothing

In the invariant framework, f_i is assumed to be group affine. These dynamics were introduced in continuous time in [5], and in discrete time in [8]. The main idea is that they extend the notion of linear dynamics (i.e. defined by affine maps) from vector spaces to Lie groups.

Definition 1. *Group affine dynamics are defined as*

$$\chi_{i+1} = f_i(\chi_i) = \Lambda_i\phi(\chi_i)\Upsilon_i. \tag{9}$$

with $\Lambda_i, \Upsilon_i \in G$, and ϕ an automorphism, i.e., satisfies (2)*.*

Group affine dynamics include a large class of systems of engineering interest revolving around navigation and robotics, as shown in e.g. [5,8,36,43]. They come with the *log-linear property*, originally introduced and proved in [5].

Proposition 1 (from [8], discrete-time log-linear property)**.** *For group affine dynamics* (9)*, we have*

$$f_i(\chi_i\exp(\xi)) = \chi_{i+1}\exp(\mathbf{F}_i\xi) \tag{10}$$

with $\mathbf{F}_i = \mathbf{Ad}_{\Upsilon_i^{-1}}\mathbf{M}$ a linear operator, and $\mathbf{M}$ from (3)*, and where $\chi_{i+1} := f_i(\chi_i)$.*

Log-linearity comes with strong properties of invariant filtering [9,14] and invariant smoothing [17], since dynamics' linearized approximation in (8) becomes exact. Moreover, they are easily shown to possess a general preintegration property, see [8,12], extending that of [24,25,35]. However the IMU equations are group affine only where sensor biases are neglected, relying on the Lie group $SE_2(3)$ [3,5,43]. The recently introduced two-frames group (TFG) structure partially overcame this limitation, and suggested a new way to account for IMU biases in a principled manner, while unifying most group-affine systems discovered so far.

III. Localization with an IMU using the TFG

This section introduces the localization problem to which we want to apply invariant smoothing using the TFG. We start by considering the accelerometer biases only.

A. Considered simplified problem

Consider a mobile body equipped with an inertial measurement unit (IMU) providing gyroscope and accelerometer measurements, and a GNSS (e.g., GPS) receiver providing position measurements $\mathbf{Y}_i$. We neglect, for now, the gyroscope bias. A simple discretization of the continuous-time equations [24] yields the discrete-time dynamics:

$$\begin{cases}\mathbf{R}_{i+1} = \mathbf{R}_i\exp_m[\Delta t(\omega_i)_\times] \\ \mathbf{v}_{i+1} = \mathbf{v}_i + \Delta t\ (\mathbf{g} + \mathbf{R}_i(\mathbf{a}_i - \mathbf{b}_i^a)) \\ \mathbf{p}_{i+1} = \mathbf{p}_i + \Delta t\ \mathbf{v}_i \\ \mathbf{b}_{i+1}^a = \mathbf{b}_i^a\end{cases}, \tag{11}$$

with observation $\mathbf{Y}_i = \mathbf{p}_i$. In the above Δt is a time step, $\mathbf{R}_i \in G = SO(3)$ denotes the transformation at time step i that maps the frame attached to the IMU (body) to the earth-fixed frame, $\mathbf{p}_i \in \mathbb{R}^3$ denotes the position of the body in space, $\mathbf{v}_i \in \mathbb{R}^3$ denotes its velocity, $\mathbf{g}$ is the earth gravity vector, $\mathbf{a}_i, \omega_i \in \mathbb{R}^3$ the accelerometer and gyroscope signals, $\mathbf{b}_i^a$ the accelerometer bias, $\exp_m$ denotes the matrix exponential, and for any vector $\beta \in \mathbb{R}^3$, the quantity $(\beta)_\times$ denotes the skew-symmetric matrix such that $(\beta)_\times \gamma = \beta \times \gamma$ for any $\gamma \in \mathbb{R}^3$.

B. Making the system group affine

To fall into the formalism of Section II, one needs to endow the state variables with a Lie group structure. Previously to the TFG theory, it was known that in the absence of IMU biases, inertial navigation is group affine [5], but how to deal with the biases was unclear. Almost all works implementing the invariant framework for inertial navigation - including the authors' - have treated IMU biases linearly, that is, completed the group structure of $SE_2(3)$, introduced in [5], with a linear structure regarding body variables (biases). The considered group composition law thus writes

$$\begin{pmatrix} \mathbf{R}_1 \\ \mathbf{v}_1 \\ \mathbf{p}_1 \\ \mathbf{b}_1^a \end{pmatrix} \bullet \begin{pmatrix} \mathbf{R}_2 \\ \mathbf{v}_2 \\ \mathbf{p}_2 \\ \mathbf{b}_2^a \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1\mathbf{R}_2 \\ \mathbf{v}_1 + \mathbf{R}_1\mathbf{v}_2 \\ \mathbf{p}_1 + \mathbf{R}_1\mathbf{p}_2 \\ \mathbf{b}_1^a + \mathbf{b}_2^a \end{pmatrix} \quad \text{(Imperfect IEKF law)} \tag{12}$$

This group law gave rise to the "Imperfect IEKF", see [3], leading to "imperfect invariant smoothing (IS)" [15,43], and led to practical successes [6,19,27]–[29,44,46]. Another possibility was also recently suggested in [23]. However, none of these approaches allowed the biased IMU equations to be group affine.

In this regard, the two-frames group structure proposed in [7] was a leap forward. Indeed, it turns out that this structure, which unifies a large number of estimation problems related to navigation, can be cast into the invariant filtering framework using the TFG group law. Following [7], the state space can be cast as a two-frame group (TFG). The TFG group composition law defines a way to combine the state variables which is defined as follows

$$\begin{pmatrix} \mathbf{R}_1 \\ \mathbf{v}_1 \\ \mathbf{p}_1 \\ \mathbf{b}_1^a \end{pmatrix} \bullet \begin{pmatrix} \mathbf{R}_2 \\ \mathbf{v}_2 \\ \mathbf{p}_2 \\ \mathbf{b}_2^a \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1\mathbf{R}_2 \\ \mathbf{v}_1 + \mathbf{R}_1\mathbf{v}_2 \\ \mathbf{p}_1 + \mathbf{R}_1\mathbf{p}_2 \\ \mathbf{b}_2^a + \mathbf{R}_2^\top \mathbf{b}_1^a \end{pmatrix} \quad \text{(TFG law)} \tag{13}$$

This defines the alternative two-frames group (TFG) law. Its identity element is $(\mathbf{I}, \mathbf{0}, \mathbf{0}, \mathbf{0})$ and the inverse is given by:

$$\begin{pmatrix} \mathbf{R} \\ \mathbf{v} \\ \mathbf{p} \\ \mathbf{b}^a \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{R}^\top \\ \mathbf{R}^\top \mathbf{v} \\ -\mathbf{R}^\top \mathbf{p} \\ -\mathbf{R}\mathbf{b}^a \end{pmatrix} \tag{14}$$

C. Theoretical results

The TFG structure enables a more principled treatment of sensors' biases, since it leads to group affine properties.

Proposition 2 ([7]). *The IMU equations in 3D with accelerometer bias are group affine, in the sense of the TFG group structure, whenever $\omega_i = \mathbf{0}$, that is, the orientation of the robot remains unchanged (but it may be arbitrary), while changes in acceleration and velocity are allowed.*

Planar motions bear an even more powerful result.

Proposition 3 ([7]). *The IMU equations in 3D with accelerometer bias are group affine, in the sense of the TFG group structure, when facing a planar vehicle, that is, all vectors are 2-dimensional and $R \in SO(2)$.*

D. Further details

We now propose a simple proof of Proposition 2, which has the merit to draw a clear connection between the present paper and the formalism recently used in robotics in [12].

Let us denote by χ the state variable, that is,

$$\chi_i := \begin{pmatrix} \mathbf{R}_i \\ \mathbf{v}_i \\ \mathbf{p}_i \\ \mathbf{b}_i^a \end{pmatrix}$$

To establish a link between the dynamics (11) and the TFG group law (13), one may write

$$\chi_{i+1} = \underbrace{\begin{pmatrix} I \\ \Delta t\mathbf{g} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}}_{:=\Lambda_i} \bullet \underbrace{\begin{pmatrix} \mathbf{R}_i \\ \mathbf{v}_i + \Delta t\mathbf{R}_i\mathbf{b}_i^a \\ \mathbf{p}_i + \Delta t\mathbf{v}_i \\ \Omega_i\mathbf{b}_i \end{pmatrix}}_{:=\phi_{\Omega_i}(\chi_i)} \bullet \underbrace{\begin{pmatrix} \Omega_i \\ \Delta t\mathbf{a}_i \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}}_{:=\Upsilon_i} \tag{15}$$

which may be proved to be strictly equivalent to (11), letting $\Omega_i := \exp_m[\Delta t(\omega_i)_\times]$. What the theory of invariant filtering says, is that, provided ϕ_{Ω_i} is an automorphism, i.e., satisfies (2), the dynamics $\chi_{i+1} = \Lambda_i \phi_{\Omega_i}(\chi_i)\Upsilon_i$, which coincides with (15), is group affine, see Definition 1. It then inherits the desirable property of being linear in the exponential coordinates, see (10), that comes with convergence properties.

We have all is needed to then verify Proposition 2. Indeed, letting $\Omega_i = I$, which corresponds indeed to $\omega_i = 0$, as $\Omega_i := \exp_m[\Delta t(\omega_i)_\times]$, we see that ϕ_I is an automorphism for the TFG law (13). Indeed, we have on the one hand

$$\phi_I\Big(\begin{pmatrix} \mathbf{R}_1 \\ \mathbf{v}_1 \\ \mathbf{p}_1 \\ \mathbf{b}_1^a \end{pmatrix} \bullet \begin{pmatrix} \mathbf{R}_2 \\ \mathbf{v}_2 \\ \mathbf{p}_2 \\ \mathbf{b}_2^a \end{pmatrix}\Big) = \begin{pmatrix} \mathbf{R}_1\mathbf{R}_2 \\ \mathbf{v}_1 + \mathbf{R}_1\mathbf{v}_2 + \Delta t\mathbf{R}_1\mathbf{R}_2(\mathbf{b}_2^a + \mathbf{R}_2^\top\mathbf{b}_1^a) \\ \mathbf{p}_1 + \mathbf{R}_1\mathbf{p}_2 + \Delta t(\mathbf{v}_1 + \mathbf{R}_1\mathbf{v}_2) \\ \mathbf{b}_2^a + \mathbf{R}_2^\top\mathbf{b}_1^a \end{pmatrix},$$

and on the other

$$\phi_I\Big(\begin{pmatrix} \mathbf{R}_1 \\ \mathbf{v}_1 \\ \mathbf{p}_1 \\ \mathbf{b}_1^a \end{pmatrix}\Big) \bullet \phi_I\Big(\begin{pmatrix} \mathbf{R}_2 \\ \mathbf{v}_2 \\ \mathbf{p}_2 \\ \mathbf{b}_2^a \end{pmatrix}\Big) = \begin{pmatrix} \mathbf{R}_1\mathbf{R}_2 \\ \mathbf{v}_1 + \Delta t\mathbf{R}_1\mathbf{b}_1^a + \mathbf{R}_1(\mathbf{v}_2 + \Delta t\mathbf{R}_2\mathbf{b}_2^a) \\ \mathbf{p}_1 + \Delta t\mathbf{v}_1 + \mathbf{R}_1(\mathbf{p}_2 + \Delta t\mathbf{v}_2) \\ \mathbf{b}_2^a + \mathbf{R}_2^\top\mathbf{b}_1^a \end{pmatrix}$$

We see both expressions coincide indeed, so that ϕ_I is an automorphism, thus proving Proposition 2.

IV. Application to biased IMU based localization

Now that we have recalled the TFG structure, we would like to leverage it for smoothing based localization, which

has never been done before. In practice, another bias needs to be considered when using an IMU, that of the gyroscope. By accounting for it explicitly, (11) then becomes

$$\begin{cases} \mathbf{R}_{i+1} = \mathbf{R}_i \exp_m [\Delta t(\omega_i - \mathbf{b}_i^\omega)_\times] \\ \mathbf{v}_{i+1} = \mathbf{v}_i + \Delta t \ (\mathbf{g} + \mathbf{R}_i (\mathbf{a}_i - \mathbf{b}_i^a)) \\ \mathbf{p}_{i+1} = \mathbf{p}_i + \Delta t \ \mathbf{v}_i \\ \mathbf{b}_{i+1}^a = \mathbf{b}_i^a, \quad \mathbf{b}_{i+1}^\omega = \mathbf{b}_i^\omega \end{cases}, \tag{16}$$

with $\mathbf{b}_i^\omega$ the gyroscope bias.

A. Computing the Jacobians

The Jacobian of (11) can be directly retrieved from [7] when $\omega_i = \mathbf{0}$ and there is no gyro bias. It can be extended to the more difficult present setting as follows. Let $\mathbf{D}_i = D_{\exp_m}(\Delta t(\omega_i - \mathbf{b}_i^\omega))$ with $D_{\exp_m}$ the differential of $\exp_m$. Then we have:

$$\mathbf{F}_i = \begin{bmatrix} \Omega_i^\top & -\Delta t \ \mathbf{D}_i(\mathbf{b}_i^\omega)_\times & & & -\Delta t \ \mathbf{D}_i \\ -\Omega_i^\top (\Delta t \ \mathbf{a}_i)_\times & & \Omega_i^\top & -\Delta t \ \Omega_i^\top & \\ & & \Delta t \ \Omega_i^\top & \Omega_i^\top & \\ (\mathbf{b}_i^a)_\times \left(\mathbf{I} - \Omega_i^\top + \Delta t \mathbf{D}_i (\mathbf{b}_i^\omega)_\times\right) & & & \mathbf{I} & \\ (\mathbf{b}_i^\omega)_\times \left(\mathbf{I} - \Omega_i^\top + \Delta t \mathbf{D}_i (\mathbf{b}_i^\omega)_\times\right) & & & & \mathbf{I} \end{bmatrix} \tag{17}$$

It coincides with the Jacobian of "imperfect IS" which relies on $SE_2(3) \times \mathbb{R}^3 \times \mathbb{R}^3$ (see [12,15]), except for the first block column, whose computation is detailed in Appendix A.

B. Differences with other parametrisations

Let us summarise the differences between IS based on the TFG structure, imperfect IS which relies on $SE_2(3) \times \mathbb{R}^3$, and the smoothing method implemented for NavState in GTSAM [21], which is a refinement of [24]. Although the considered residuals and their covariances are essentially the same, the main difference lies in the parametrisation of the state (i.e. the retraction) used to update the state variables at each optimization descent step. Obviously, the treatment of the bias $\mathbf{b} = (\mathbf{b}^a, \mathbf{b}^\omega)$ is a key difference, since all other parametrisations consider it linearly. As concerns the navigational part (attitude, velocity, position), only imperfect IS coincides with IS. More precisely, the retractions used in GTSAM [21] and imperfect IS are respectively

$$(\hat{\mathbf{R}}, \hat{\mathbf{v}}, \hat{\mathbf{x}}, \mathbf{b}) \leftarrow (\hat{\mathbf{R}}\delta_R, \hat{\mathbf{v}} + \hat{\mathbf{R}}\delta_v, \hat{\mathbf{x}} + \hat{\mathbf{R}}\delta_x, \mathbf{b} + \delta_\mathbf{b}). \tag{18}$$

$$(\hat{\mathbf{R}}, \hat{\mathbf{v}}, \hat{\mathbf{x}}, \mathbf{b}) \leftarrow ((\hat{\mathbf{R}}, \hat{\mathbf{v}}, \hat{\mathbf{x}}) \exp_{SE_2(3)}(\delta_R, \delta_v, \delta_x), \mathbf{b} + \delta_\mathbf{b}). \tag{19}$$

(18) is linear by nature whereas imperfect IS (19) uses the exponential map of $SE_2(3)$, which offers a more appropriate nonlinear map. Note that (18) is a first-order approximation of (19). By constrast, the method proposed herein proposed the full exponential of the TFG, see (23) below.

The Jacobian for GTSAM (18) is similar to the one of imperfect IS, replacing $\Omega_i^\top$ and $\mathbf{a}_i$ with their estimated counterparts: $\hat{\Omega}_i^\top = \hat{\mathbf{R}}_{i+1}^\top \hat{\mathbf{R}}_i$ and $\Delta t \ \hat{\mathbf{a}}_i = \hat{\mathbf{R}}_i^\top (\hat{\mathbf{v}}_{i+1} - \hat{\mathbf{v}}_i - \Delta t \ \mathbf{g}) - \hat{\mathbf{b}}_i^a$. Though, it differs more starkly with that proposed herein.

V. Experimental Results

We evaluate the proposed smoothing method, based on the TFG, on the KITTI dataset [26], which contains raw and synchronised IMU data, and a ground-truth. Smoothing estimates are implemented using sliding windows, where the oldest state is marginalised once a given size is reached. TFG is compared with the two other parametrisations (18), (19), based on a common smoothing implementation to ensure fair comparison.

When a robot is started, typically a terrestrial wheeled vehicle, it may have no information about its position and its orientation. However, in outdoors applications, GNSS provides a quick coarse estimate of the position, and the accelerometers that of the vertical direction. The yaw, on the other hand, is much harder to estimate, as no coarse estimate is directly available (magnetometers may be very unreliable due to metallic masses in vehicles and robots). To highlight the differences between the various smoothing methods, we consider the difficult and relevant localization problem where initial orientation is unknown and thus orientation error may be large (100 ° standard deviation), and we focus on the transitory phase during which orientation is recovered (this is called the alignment problem in aerospace engineering, see e.g., [20,37,47]).

A. Implementation details

The IMU data is sampled at 100Hz. We simulate noisy position measurements at 1 Hz by adding white noise to the ground truth with $\sigma_y = 1m$, which is typically the accuracy of a GNSS sensor.

The estimators are compared using three different window sizes of 5, 10 and 15 instants for the state. The initial position is based on the ground truth, roll, pitch, velocity and biases are initialised at zero, and the yaw is randomly sampled. The initial uncertainty is given by $\sigma_p^0 = 1$m, $\sigma_v^0 = 10$m/s, $\sigma_R^0 = 100$ °, $\sigma_{b^a}^0 = 0.06$m/s^2, $\sigma_{b^\omega}^0 = 0.07$rad/s. The process uncertainty uses $\sigma_a = 0.05$m/s^2, $\sigma_\omega = 0.01$rad/s, $\sigma_{b^a} = 0.002$m/s^2, $\sigma_{b^\omega} = 3.10^{-5}$rad/s. For each experiment, 50 Monte Carlo runs are carried out.

B. Results

The kind of considered "robot kidnapping" problem, where the robot has no initial prior information, is difficult. Indeed, the original heading error may be large, so that the estimate can fall into a wrong local minima, and be slow or even fail to converge. Therefore, we compare the estimators in terms of consistency: We consider the estimator to be consistent if its yaw error stays inside its believed 3σ envelope. Consistency is thus defined as the ability to convey a consisent (or conservative) estimate of the extent of uncertainty associated with the predictions. For collision avoidance and online motion planning, it is important indeed to be able to correctly (or conservatively) assess the errors.

Figure 1 shows the yaw errors over time for the three smoothing methods, highlighting trajectories which yield inconsistent estimates for each of the 50 random initializations, over a typical KITTI trajectory example. We observe a much larger ratio of orientation errors fail to stay inside the 3σ envelope when using alternative methods to the TFG-based one proposed in the present paper. Moreover, we see that, when using the TFG, the envelopes are less dispersed over the Monte Carlo runs, indicating a reduced dependency of

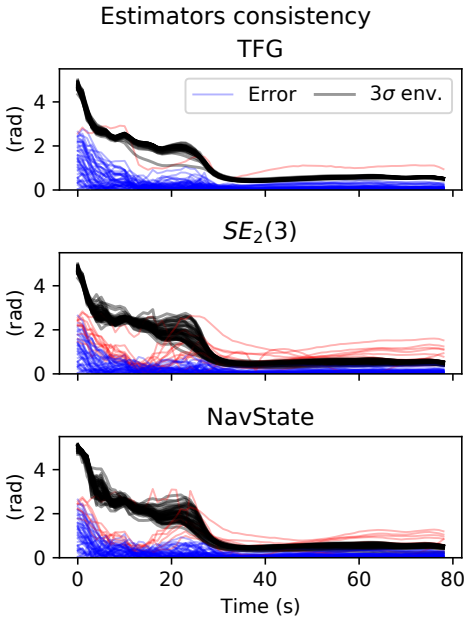

Fig. 1: Convergence and consistency comparison of smoothing based on three parametrisations: using the TFG (ours), $SE_2(3)$ [17], and NavState (GTSAM [21]), with a window size of 5. For 50 Monte Carlo runs over the KITTI trajectory 01, during the transitory phase after an initialization with random yaw, the yaw error and the 3σ envelope obtained from the estimated covariance are plotted over time. Consistent trajectories are in blue, inconsistent ones in red. We see that the proposed method provides a greater ratio of consistent estimates. Besides, we note the 3σ envelopes are less dispersed over Monte-Carlo runs when using the TFG, which is consistent with them having less dependency on the estimates, see [7].

the covariance matrix on the estimated trajectory, which is the main goal of the TFG theory indeed [7].

To assess the method over a larger number of experiments, Table I reports, for the various KITTI trajectories, window sizes and methods, the ratio of Monte Carlo runs which are consistent. As expected, larger windows ensure better convergence (but require more onboard computation capabilities) except when the ratio is already close to one. Moreover, smoothing based on the TFG proves much more robust, in that it systematically outperforms other methods when using the smallest window size (and hence being computationally efficient). It indicates a special ability to avoid local minima.

VI. Conclusion

In this paper we proposed to leverage the recently introduced Two-Frames Group structure in an Invariant Smoothing framework for localization, in a difficult setting where the window size is relatively low, and the inital error may be large. The necessary derivations, notably to retrieve the Jacobians, helped understanding the properties of the TFG. The proposed method showed increased robustness over the state-of-the-art in terms of convergence on real data.

seq.	window	TFG	$SE_2(3)$	(18) (GTSAM)
01	5	**0.98**	0.78	0.86
	10	**0.98**	**0.98**	0.96
	15	**0.98**	**0.98**	0.84
05	5	**0.94**	0.7	0.7
	10	**0.96**	0.92	0.92
	15	0.96	**0.98**	0.96
06	5	**0.96**	0.68	0.66
	10	0.94	0.96	**1**
	15	0.94	0.98	**1**
07	5	**0.96**	0.94	**0.96**
	10	**0.98**	**0.98**	**0.98**
	15	0.98	**1**	**1**
08	5	**0.98**	0.84	0.86
	10	**0.98**	**0.98**	**0.98**
	15	**1**	0.98	0.94
09	5	**0.84**	0.68	0.7
	10	**0.94**	0.86	0.88
	15	0.96	**1**	0.96
10	5	**0.96**	0.88	0.88
	10	**1**	0.96	0.98
	15	**1**	0.96	0.94

TABLE I: Ratio of consistent trajectories, out of 50 random Monte Carlo initialization, for various smoothing methods on the KITTI dataset. Consistency is defined as the yaw error staying inside the 3-σ envelope.

This supports the relevance of the TFG for more complex navigation problems, which opens up for a wide range of future work directions. It may be interesting in the future to investigate notbaly whether one could use the framework of GTSAM combined with this new kind of Lie-group based parameterization instead of the current on-manifold parametrization, as to date only improvements have been observed. Further investigations are thus desirable.

Appendix A
Computation of (17)

Let $\chi = (R, \mathbf{v}, \mathbf{p}, \mathbf{b}^a, \mathbf{b}^\omega)$ be the state. For readability, we denote $\exp(\theta) = \exp_m[(\theta)_\times]$ on $SO(3)$, i.e. for $\theta \in \mathbb{R}^3$. Consider an update $\delta_\chi = \begin{pmatrix}\delta_\mathbf{R} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{pmatrix}$, with $\delta_\mathbf{R} = \exp(\xi^R)$. Then we have $\chi \bullet \delta_\chi = (R\delta_\mathbf{R}, \mathbf{v}, \mathbf{p}, \delta_\mathbf{R}^\top \mathbf{b}^a, \delta_\mathbf{R}^\top \mathbf{b}^\omega)$, and

$$f_i(\chi \bullet \delta_\chi) = \begin{pmatrix} \mathbf{R}\delta_\mathbf{R} \exp(\Delta t(\omega_i - \delta_\mathbf{R}^\top \mathbf{b}^\omega)) \\ \mathbf{v} + \Delta t\ (\mathbf{g} + \mathbf{R}\delta_\mathbf{R}(\mathbf{a}_i - \mathbf{b}^a)) \\ \mathbf{p} + \Delta t\ \mathbf{v} \\ \delta_\mathbf{R}^\top \mathbf{b}^a \\ \delta_\mathbf{R}^\top \mathbf{b}^\omega \end{pmatrix} \tag{20}$$

Let $\Omega_i = \exp(\Delta t(\omega_i - \mathbf{b}^\omega))$. Since $\delta_\mathbf{R}^\top \approx \mathbf{I} - (\xi^R)_\times$, we can approximate $\exp(\Delta t(\omega_i - \delta_\mathbf{R}^\top \mathbf{b}^\omega)) \approx \Omega_i \exp(-\Delta t \mathbf{D}_i (\xi^R)_\times \mathbf{b}^\omega)$ with $\mathbf{D}_i = D_{\exp}(\Delta t(\omega_i - \mathbf{b}^\omega))$ and $D_{\exp}$ the differential of exp on $SO(3)$. We know that $\mathbf{R}\delta_\mathbf{R}\Omega_i = \mathbf{R}\Omega_i \exp(\Omega_i^\top \xi^R)$, so $\mathbf{R}\delta_\mathbf{R} \exp(\Delta t(\omega_i - \delta_\mathbf{R}^\top \mathbf{b}^\omega)) \approx \mathbf{R}\Omega_i \exp(\Omega_i^\top \xi^R - \Delta t \mathbf{D}_i (\mathbf{b}^\omega)_\times \xi^R)$. Therefore, given (13), we

can write

$$f_i(\chi \bullet \delta_\chi) \approx f_i(\chi) \bullet \begin{pmatrix} \exp(\Omega_i^\top \xi^R - \Delta t \mathbf{D}_i(\mathbf{b}^\omega)_\times \xi^R) \\ -\Delta t \Omega_i^\top (\xi^R)_\times \mathbf{a}_i \\ 0 \\ \delta_\mathbf{R}^\top \mathbf{b}^a - \exp(-\Omega_i^\top \xi^R + \Delta t \mathbf{D}_i(\mathbf{b}^\omega)_\times \xi^R)\mathbf{b}^a \\ \delta_\mathbf{R}^\top \mathbf{b}^\omega - \exp(-\Omega_i^\top \xi^R + \Delta t \mathbf{D}_i(\mathbf{b}^\omega)_\times \xi^R)\mathbf{b}^\omega \end{pmatrix} \tag{21}$$

The last terms usually do not appear when considering other parametrisations. Linearising the exponential, we get

$$\begin{aligned} &\left(\delta_\mathbf{R}^\top - \exp(-\Omega_i^\top \xi^R + \Delta t \mathbf{D}_i(\mathbf{b}^\omega)_\times \xi^R)\right)\mathbf{b}^a \\ &\approx \left(-(\xi^R)_\times + (\Omega^\top \xi^R)_\times - \Delta t (\mathbf{D}_i(\mathbf{b}^\omega)_\times \xi^R)_\times\right)\mathbf{b}^a \\ &= (\mathbf{b}^a)_\times \left(\mathbf{I} - \Omega^\top + \Delta t \mathbf{D}_i(\mathbf{b}^\omega)_\times\right)\xi^R \end{aligned} \tag{22}$$

and similarly for $\mathbf{b}^\omega$, thus recovering the first block column of (17). In particular, this highlights why $\Omega_i \neq Id$ breaks the group affine property of (11) in the 3D case (and without the gyro bias).

APPENDIX B
USEFUL FORMULAS FOR THE TFG

We recall the results of [7] being useful herein.

A. Exponential and Logarithm

The exponential on the considered TFG is given by the following formula:

$$\exp_{TFG}\begin{pmatrix} \xi^R \\ \xi^v \\ \xi^p \\ \xi^{b^a} \\ \xi^{b^\omega} \end{pmatrix} = \begin{pmatrix} \exp(\xi^R) \\ \nu(\xi^R)\xi^v \\ \nu(\xi^R)\xi^p \\ \nu(-\xi^R)\xi^{b^a} \\ \nu(-\xi^R)\xi^{b^\omega} \end{pmatrix} \tag{23}$$

Where ν is given by:

$$\nu(\xi) = \mathbf{I} + \frac{1-\cos(||\xi||)}{||\xi||^2}(\xi)_\times + \frac{||\xi|| - \sin(||\xi||)}{||\xi||^3}(\xi)_\times^2$$

Its inverse, the logarithm, writes:

$$\log_{TFG}\begin{pmatrix} \mathbf{R} \\ \mathbf{v} \\ \mathbf{p} \\ \mathbf{b}^a \\ \mathbf{b}^\omega \end{pmatrix} = \begin{pmatrix} \xi^R \\ \nu(\xi^R)^{-1}\mathbf{v} \\ \nu(\xi^R)^{-1}\mathbf{p} \\ \nu(-\xi^R)^{-1}\mathbf{b}^a \\ \nu(-\xi^R)^{-1}\mathbf{b}^\omega \end{pmatrix} \tag{24}$$

B. Adjoint Matrices and left Jacobian

In order to apply (8) to the TFG, one needs to compute $\widetilde{\mathbf{P}}_0 = \mathbf{J}_0^{-1}\mathbf{P}_0\mathbf{J}_0^{-T}$, where $\mathbf{J}_0$ is the left Jacobian of the group computed at $\mathbf{p}_0$, which is given by the following sum $\mathbf{J}_0 = \sum_{j\geq 0}\frac{1}{(n+1)!}(\mathbf{ad}_{\mathbf{p}_0})^j$ [1]. Moreover $\mathbf{ad}_{\mathbf{p}_0}$ can be determined through identification, thanks to the fact that $\exp_m \mathbf{ad}_{\mathbf{p}_0} = \mathbf{Ad}_{\exp_{TFG}\mathbf{p}_0}$. Thus, what is left to do is compute $\mathbf{Ad}_\chi$ for an element of the TFG. In general it is given by the differential of $g \mapsto \chi \bullet g \bullet \chi^{-1}$ around the identity. Let $\chi = (\mathbf{R}, \mathbf{v}, \mathbf{p}, \mathbf{b}^a, \mathbf{b}^\omega)$ be a state. Then, following [7], we get

$$\mathbf{Ad}_\chi = \begin{bmatrix} \mathbf{R} & & & & & \\ (\mathbf{v})_\times\mathbf{R} & \mathbf{R} & & & & \\ (\mathbf{p})_\times\mathbf{R} & & \mathbf{R} & & & \\ \mathbf{R}(\mathbf{b}^a)_\times & & & \mathbf{R} & & \\ \mathbf{R}(\mathbf{b}^\omega)_\times & & & & \mathbf{R} & \end{bmatrix} \tag{25}$$

Now, let $\chi = \exp_{TFG}(\xi)$, with $\xi = (\xi^R, \xi^p, \xi^p, \xi^{b^a}, \xi^{b^\omega})$. Using (23), and the facts that $\mathbf{R}\nu(-\xi^R) = \nu(\xi^R)$, and $\mathbf{R}(\mathbf{x})_\times = (\mathbf{R}\mathbf{x})_\times\mathbf{R}$, we get

$$\mathbf{Ad}_{\exp_{TFG}(\xi)} = \begin{bmatrix} \mathbf{R} & & & & & \\ (\nu(\xi^R)\xi^v)_\times\mathbf{R} & \mathbf{R} & & & & \\ (\nu(\xi^R)\xi^p)_\times\mathbf{R} & & \mathbf{R} & & & \\ (\nu(\xi^R)\xi^{b^a})_\times\mathbf{R} & & & \mathbf{R} & & \\ (\nu(\xi^R)\xi^{b^\omega})_\times\mathbf{R} & & & & \mathbf{R} & \end{bmatrix} \tag{26}$$

Therefore, we can directly identify

$$\mathbf{ad}_\xi = \begin{bmatrix} (\xi^R)_\times & & & & & \\ (\xi^v)_\times & (\xi^R)_\times & & & & \\ (\xi^p)_\times & & (\xi^R)_\times & & & \\ (\xi^{b^a})_\times & & & (\xi^R)_\times & & \\ (\xi^{b^\omega})_\times & & & & (\xi^R)_\times & \end{bmatrix} \tag{27}$$

REFERENCES

[1] Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[2] Timothy D. Barfoot and Paul. T. Furgale. Associating Uncertainty with Three-Dimensional Poses for Use in Estimation Problems. *IEEE Transactions on Robotics*, 30(3):679–693, 2014.

[3] Axel Barrau. *Non-linear state error based extended Kalman filters with applications to navigation*. PhD thesis, Mines Paristech, 2015.

[4] Axel Barrau and Silvère Bonnabel. An EKF-SLAM algorithm with consistency properties. *CoRR*, abs/1510.06263, 2015.

[5] Axel Barrau and Silvère Bonnabel. The invariant extended kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, 62(4):1797–1812, April 2017.

[6] Axel Barrau and Silvere Bonnabel. Invariant kalman filtering. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:237–257, 2018.

[7] Axel Barrau and Silvere Bonnabel. The geometry of navigation problems. *IEEE Transactions on Automatic Control*, 68(2):689–704, 2022.

[8] Axel Barrau and Silvère Bonnabel. Linear observed systems on groups. *Systems and Control Letters*, 129:36 – 42, 2019.

[9] Axel Barrau and Silvère Bonnabel. Extended kalman filtering with nonlinear equality constraints: A geometric approach. *IEEE Transactions on Automatic Control*, 65(6):2325–2338, 2020.

[10] Guillaume Bourmaud, Rémi Mégret, Audrey Giremus, and Yannick Berthoumieu. Discrete extended Kalman filter on Lie groups. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, pages 1–5. IEEE, 2013.

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

[12] Martin Brossard, Axel Barrau, Paul Chauchat, and Silvère Bonnabel. Associating uncertainty to extended poses for on Lie group IMU preintegration with rotating Earth. *IEEE Transactions on Robotics*, 38(2):998–1015, 2021.

[13] D. Caruso, A. Eudes, M. Sanfourche, D. Vissière, and G. L. Besnerais. Magneto-visual-inertial dead-reckoning: Improving estimation consistency by invariance. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7923–7930, 2019.

[14] P. Chauchat, A. Barrau, and S. Bonnabel. Kalman filtering with a class of geometric state equality constraints. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2581–2586, Dec 2017.

[15] Paul Chauchat. *Smoothing algorithms for navigation, localisation and mapping based on high-grade inertial sensors*. Theses, Université Paris sciences et lettres, February 2020.

[16] Paul Chauchat, Axel Barrau, and Silvere Bonnabel. Invariant Smoothing on Lie Groups. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018*, Madrid, Spain, October 2018.

[17] Paul Chauchat, Silvére Bonnabel, and Axel Barrau. Invariant smoothing with low process noise. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 4758–4763, 2022.

[18] Gregory S Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011.

[19] Mitchell R Cohen and James Richard Forbes. Navigation and control of unconventional vtol uavs in forward-flight with explicit wind velocity estimation. *IEEE Robotics and Automation Letters*, 5(2):1151–1158, 2020.

[20] Xiao Cui, Chunbo Mei, Yongyuan Qin, Gongmin Yan, and Qiangwen Fu. In-motion alignment for low-cost sins/gps under random misalignment angles. *Journal of Navigation*, 70(6):1224–1240, 2017.

[21] Franck Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.

[22] Frank Dellaert, Michael Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017.

[23] Alessandro Fornasier, Yonhon Ng, Robert Mahony, and Stephan Weiss. Equivariant filter design for inertial navigation systems with input measurement biases. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4333–4339. IEEE, 2022.

[24] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, Feb 2017.

[25] Médéric Fourmy, Thomas Flayols, Pierre-Alexandre Léziart, Nicolas Mansard, and Joan Solà. Contact forces preintegration for estimation in legged robotics using factor graphs. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1372–1378. IEEE, 2021.

[26] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[27] Ross Hartley, Maani Ghaffari, Ryan M Eustice, and Jessy W Grizzle. Contact-aided invariant extended kalman filtering for robot state estimation. *The International Journal of Robotics Research*, 39(4):402–430, 2020.

[28] Ross Hartley, Maani Ghaffari Jadidi, Jessy Grizzle, and Ryan M Eustice. Contact-aided invariant extended kalman filtering for legged robot state estimation. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[29] S. Heo and C. G. Park. Consistent ekf-based visual-inertial odometry on matrix lie group. *IEEE Sensors Journal*, 18(9):3780–3788, May 2018.

[30] Minh-Duc Hua, Guillaume Ducard, Tarek Hamel, Robert Mahony, and Konrad Rudin. Implementation of a nonlinear attitude estimator for aerial robotic vehicles. *Control Systems Technology, IEEE Transactions on*, 22(1):201–213, 2014.

[31] V. Indelman, S. Williams, Michael Kaess, and F. Dellaert. Information fusion in navigation systems via factor graph based incremental smoothing. *Journal of Robotics and Autonomous Systems, RAS*, 61(8):721–738, August 2013.

[32] Joon-Ha Kim, Seungwoo Hong, Gwanghyeon Ji, Seunghun Jeon, Jemin Hwangbo, Jun-Ho Oh, and Hae-Won Park. Legged robot state estimation with dynamic contact event information. *IEEE Robotics and Automation Letters*, 6(4):6733–6740, 2021.

[33] Haomin Liu, Mingyu Chen, Guofeng Zhang, Hujun Bao, and Yingze Bao. Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[34] Andrew W Long, Kevin C Wolfe, Michael J Mashner, Gregory S Chirikjian, et al. The banana distribution is gaussian: A localization study with exponential coordinates. *Robotics: Science and Systems VIII*, 265:1, 2013.

[35] T. Lupton and S. Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1):61–76, Feb 2012.

[36] Robert E. Mahony and Tarek Hamel. A geometric nonlinear observer for simultaneous localisation and mapping. In *56th IEEE Annual Conference on Decision and Control, CDC 2017, Melbourne, Australia, December 12-15, 2017*, pages 2408–2415, 2017.

[37] Wei Ouyang and Yuanxin Wu. Optimization-based strapdown attitude alignment for high-accuracy systems: Covariance analysis with applications. *IEEE Transactions on Aerospace and Electronic Systems*, 2022.

[38] Wooram Park, Yan Liu, Yu Zhou, Matthew Moses, Gregory S. Chirikjian, et al. Kinematic state estimation and motion planning for stochastic nonholonomic systems using the exponential map. *Robotica*, 26(4):419–434, 2008.

[39] Natalia Pavlasek, Alex Walsh, and James Richard Forbes. Invariant extended kalman filtering using two position receivers for extended pose estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5582–5588. IEEE, 2021.

[40] Joan Sola, Jeremie Deray, and Dinesh Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018.

[41] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[42] Niels van Der Laan, Mitchell Cohen, Jonathan Arsenault, and James Richard Forbes. The invariant rauch-tung-striebel smoother. *IEEE Robotics and Automation Letters*, 5(4):5067–5074, 2020.

[43] A. Walsh, J. Arsenault, and J. R. Forbes. Invariant sliding window filtering for attitude and bias estimation. In *2019 American Control Conference (ACC)*, pages 3161–3166, July 2019.

[44] Miaomiao Wang and Abdelhamid Tayebi. Hybrid nonlinear observers for inertial navigation using landmark measurements. *IEEE Transactions on Automatic Control*, 65(12):5173–5188, 2020.

[45] Kevin C. Wolfe, Michael Mashner, and Gregory S. Chirikjian. Bayesian fusion on Lie groups. *Journal of Algebraic Statistics*, 2(1):75–97, 2011.

[46] K. Wu, T. Zhang, D. Su, S. Huang, and G. Dissanayake. An invariant-ekf vins algorithm for improving consistency. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1578–1585, Sep. 2017.

[47] Yuanxin Wu and Xianfei Pan. Velocity/position integration formula part i: Application to in-flight coarse alignment. *IEEE Transactions on Aerospace and Electronic Systems*, 49(2):1006–1023, APRIL 2013.

[48] Sheng Zhao, Yiming Chen, Haiyu Zhang, and Jay A. Farrell. Differential gps aided inertial navigation: a contemplative realtime approach. *IFAC Proceedings Volumes*, 47(3):8959 – 8964, 2014. 19th IFAC World Congress.