

This paper has been accepted for publication in *IEEE Robotics and Automation Letters*.

This is the author's version of an article that has, or will be, published in this journal or conference.
Changes were, or will be, made to this version by the publisher prior to publication.

DOI: 10.1109/LRA.2020.3005132
IEEE Xplore: <https://ieeexplore.ieee.org/document/9126191>

Please cite this paper as:

N. van der Laan, M. Cohen, J. Arsenault, and J. R. Forbes, "The Invariant Rauch-Tung-Striebel Smoother," in *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5067-5074, June 2020.

arXiv:2403.00075v1 [cs.RO] 29 Feb 2024

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

The Invariant Rauch-Tung-Striebel Smoother

Niels van der Laan¹, Mitchell Cohen², Jonathan Arsenault³ and James Richard Forbes⁴

Abstract—This paper presents an invariant Rauch-Tung-Striebel (IRTS) smoother applicable to systems with states that are an element of a matrix Lie group. In particular, the extended Rauch-Tung-Striebel (RTS) smoother is adapted to work within a matrix Lie group framework. The main advantage of the invariant RTS (IRTS) smoother is that the linearization of the process and measurement models is independent of the state estimate resulting in state-estimate-independent Jacobians when certain technical requirements are met. A sample problem is considered that involves estimation of the three dimensional pose of a rigid body on $SE(3)$, along with sensor biases. The multiplicative RTS (MRTS) smoother is also reviewed and is used as a direct comparison to the proposed IRTS smoother using experimental data. Both smoothing methods are also compared to invariant and multiplicative versions of the Gauss-Newton approach to solving the batch state estimation problem.

Index Terms—Localization, autonomous vehicle navigation, sensor fusion.

I. INTRODUCTION

The need to estimate states using incomplete and noisy data arises in many engineering problems. In robotics applications, where states of interest are often elements of a matrix Lie group, a popular state estimator is the extended Kalman filter (EKF), or rather the multiplicative extended Kalman filter (MEKF), a variant of the EKF [1]. The EKF is an approximation to the Bayes filter that uses linearization to compute a state estimate [2, Sec. 4.2]. The prediction and correction steps of the EKF use process and measurement model Jacobians, respectively, that are evaluated at the most recent state estimate. If the most recent state estimate is poor, the Jacobians are inaccurate.

The invariant EKF (IEKF) is a state estimator that is specific to systems with states that are an element of a matrix Lie group. It was shown in [3], [4] that when the process model is group affine, the measurements are left invariant (right invariant), and a left-invariant error (right-invariant error) is used,

the IEKF as interpreted as an observer possesses asymptotic convergence properties. Moreover, the Jacobians associated with the IEKF, when treated as an observer, are state-estimate independent. As such, the IEKF has enhanced performance properties when compared to the EKF or MEKF [3]–[5].

Unlike the Kalman filter and its nonlinear variants, the Rauch-Tung-Striebel (RTS) smoother uses all available measurements in a batch estimation framework to compute state estimates using a forward pass followed by a backward pass. In [6] an RTS smoother for systems with states that are an element of a matrix Lie group is presented, but the invariant framework is not leveraged. Batch estimation in an invariant framework is considered in [7], with additional details and extensions given in [8], leading to an invariant Gauss-Newton (IGN) algorithm. Invariant sliding window filtering is also considered in [9] in the context of attitude and bias estimation. Applying the invariant framework to the simultaneous localization and mapping problem in a batch framework is considered in [5]. The advantage of the invariant framework is the fact that the Jacobians are state-estimate independent. Motivated by this fact, this paper considers the derivation of an invariant RTS (IRTS) smoother for systems with states that are elements of a matrix Lie group. Like the IEKF and IGN, the IRTS smoother has state-independent Jacobians when certain technical requirements are met.

To assess the performance of the IRTS smoother relative to a multiplicative RTS (MRTS) smoother, IGN, as well as multiplicative GN (MGN), a state estimation problem is considered that involves estimating the position and attitude of a rigid body. Two interoceptive sensor measurements, as well as two exteroceptive measurements, are available. Angular velocity and translational velocity measurements are the interoceptive measurements, while both left-invariant and right-invariant measurements from a GPS receiver and a stereo camera are the exteroceptive measurements. In addition, biases on the angular and translational velocity sensors are also estimated. Due to the inclusion of sensor biases, the state estimation problem does not satisfy the exact requirements of the invariant framework, that being a group-affine process model. However, leveraging the invariant framework still leads to Jacobians that depend less on the state estimates as compared to the multiplicative framework [5]. The IRTS smoother, MRTS smoother, IGN, and MGN are tested and compared using the Starry Night dataset [10]. The IRTS smoother outperforms MRTS smoothing, IGN and MGN when initial errors are large. The enhanced performance of the IRTS smoother stems from the fact that the Jacobians associated with the IRTS smoother are less state-estimate-dependent than those used in MRTS smoothing, IGN and MGN, and during the forward-backward passes of the IRTS smoother the Jacobians are updated at each step.

Manuscript received: February 24th, 2020; Revised May 13th, 2020; Accepted June 8th, 2020.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Group for Research in Decision Analysis (GERAD) and the National Science and Engineering Research Council (NSERC) of Canada.

¹Research Trainee, Group for Research in Decision Analysis (GERAD), Montreal, QC, H3T 1J4, Canada. Email: n.vanderlaan-2@student.tudelft.nl

²M.Eng. Candidate, Department of Mechanical Engineering, McGill University, Montreal, QC, H3A 0C3, Canada. Email: mitchell.cohen3@mail.mcgill.ca

³Ph.D. Candidate, Department of Mechanical Engineering, McGill University, Montreal, QC, H3A 0C3, Canada. Email: jonathan.arsenault@mail.mcgill.ca

⁴Associate Professor, Department of Mechanical Engineering, McGill University, Montreal, QC, H3A 0C3, Canada. Email: james.richard.forbes@mcgill.ca

Digital Object Identifier (DOI): see top of this page.

The remainder of this paper is organized as follows. Notation and preliminaries are presented in Section II. In Section III the standard RTS smoother is reviewed. An extended RTS smoother applicable to systems with states that are an element of matrix Lie group, the MRTS smoother, is presented in Section IV. Finally, the invariant framework is brought to bear on the RTS smoothing problem in Section V. Section VI formulates the $SE(3)$ with bias state estimation problem, while section VII presents experimental results comparing the IRTS and MRTS smoothers, as well as IGN and MGN. The paper is drawn to a close in Section VIII. This paper's contributions are the IRTS smoother derivation in Section V and its evaluation and comparison to three other batch estimation methods using experimental data in Section VII.

II. PRELIMINARIES

A. Matrix Lie Groups

Consider the matrix Lie group \mathcal{G} , which is composed of $n \times n$ matrices with m degrees of freedom, that is closed under matrix multiplication [11]. The matrix Lie algebra associated with \mathcal{G} , denoted by \mathfrak{g} , is the tangent space of \mathcal{G} at the identity element $\mathbf{1}$, denoted as $T_1\mathcal{G}$. The matrix Lie algebra can be mapped to the matrix Lie group using the exponential map, $\exp(\cdot) : \mathfrak{g} \rightarrow \mathcal{G}$. The inverse map uses the matrix natural logarithm, $\log(\cdot) : \mathcal{G} \rightarrow \mathfrak{g}$. The linear operator $(\cdot)^\wedge : \mathbb{R}^m \rightarrow \mathfrak{g}$ maps an m dimension column matrix to the matrix Lie algebra. The inverse map is $(\cdot)^\vee : \mathfrak{g} \rightarrow \mathbb{R}^m$. Let $\text{Ad}(\cdot)$ denote the matrix representation of the adjoint operator, where $(\text{Ad}(\mathbf{X})\xi)^\wedge = \mathbf{X}\xi^\wedge\mathbf{X}^{-1}$ for $\mathbf{X} \in \mathcal{G}$ and $\xi \in \mathbb{R}^m$.

Definition 1 (Group Affine [3]): The function $\mathbf{F}(\mathbf{X}, \mathbf{u})$ is said to be group affine if for $\mathbf{X}_1, \mathbf{X}_2 \in \mathcal{G}$ it satisfies

$$\mathbf{F}(\mathbf{X}_1\mathbf{X}_2, \mathbf{u}) = \mathbf{X}_1\mathbf{F}(\mathbf{X}_2, \mathbf{u}) + \mathbf{F}(\mathbf{X}_1, \mathbf{u})\mathbf{X}_2 - \mathbf{X}_1\mathbf{F}(\mathbf{1}, \mathbf{u})\mathbf{X}_2. \quad (1)$$

Definition 2 (Left- and Right-Invariant Error [3]): Consider $\mathbf{X}, \hat{\mathbf{X}} \in \mathcal{G}$. The left-invariant error between \mathbf{X} and $\hat{\mathbf{X}}$ is given by $\delta\mathbf{X}^L = \mathbf{X}^{-1}\hat{\mathbf{X}}$ and the right-invariant error between \mathbf{X} and $\hat{\mathbf{X}}$ is $\delta\mathbf{X}^R = \hat{\mathbf{X}}\mathbf{X}^{-1}$.

III. RTS SMOOTHING

Consider the linear system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}\mathbf{w}(t), \quad (2)$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{M}_k\mathbf{v}_k, \quad (3)$$

where k denotes the time steps such that $\mathbf{x}_k = \mathbf{x}(t_k)$, $\mathbf{x}(t) \in \mathbb{R}^m$ is the state, $\mathbf{u}(t)$ is an interceptive measurement, \mathbf{y}_k is the exteroceptive measurement, $\mathbf{w}(t)$ is process noise, and \mathbf{v}_k is measurement noise. Unless required for clarity, the argument (t) will be omitted for brevity. The discrete-time process model can be found in several ways [12] resulting in the discrete-time process model

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{L}_{k-1}\mathbf{w}_{k-1}. \quad (4)$$

The RTS smoother consists of a forward pass, equivalent to the Kalman filter, that goes forward in time and a backward smoothing pass that goes backward in time.

The predicted state estimate and covariance are found using [13, Sec. 8.2]

$$\check{\mathbf{x}}_{f,k} = \mathbf{A}_{k-1}\hat{\mathbf{x}}_{f,k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1}, \quad (5)$$

$$\check{\mathbf{P}}_{f,k} = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{f,k-1}\mathbf{A}_{k-1}^\top + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^\top, \quad (6)$$

where $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$, $\check{\mathbf{x}}_{f,k}$ is the *a priori* forward state estimate, $\hat{\mathbf{x}}_{f,k}$ is the *a posteriori* forward state estimate, $\check{\mathbf{P}}_{f,k}$ is the *a priori* forward covariance estimate, $\hat{\mathbf{P}}_{f,k}$ is the *a posteriori* forward covariance estimate, and $k = 0, \dots, N$. The Kalman gain $\mathbf{K}_{f,k}$ is computed using [13, Sec. 8.2]

$$\mathbf{K}_{f,k} = \check{\mathbf{P}}_{f,k}\mathbf{H}_k^\top (\mathbf{H}_k\check{\mathbf{P}}_{f,k}\mathbf{H}_k^\top + \mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^\top)^{-1}, \quad (7)$$

where $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. The predicted estimates are then corrected using [14, Sec. 9.4]

$$\hat{\mathbf{x}}_{f,k} = \check{\mathbf{x}}_{f,k} + \mathbf{K}_{f,k}(\mathbf{y}_k - \mathbf{H}_k\check{\mathbf{x}}_{f,k}), \quad (8)$$

$$\hat{\mathbf{P}}_{f,k} = (\mathbf{1} - \mathbf{K}_{f,k}\mathbf{H}_k)\check{\mathbf{P}}_{f,k}(\mathbf{1} - \mathbf{K}_{f,k}\mathbf{H}_k)^\top + \mathbf{K}_{f,k}\mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^\top\mathbf{K}_{f,k}^\top. \quad (9)$$

After the forward pass is complete, the backward smoothing pass is initialized using $\hat{\mathbf{x}}_{f,N}$ and $\hat{\mathbf{P}}_{f,N}$. The smoothing equations [13, Sec. 8.2]

$$\mathbf{K}_{s,k} = \hat{\mathbf{P}}_{f,k}\mathbf{A}_k^\top\check{\mathbf{P}}_{f,k+1}^{-1}, \quad (10)$$

$$\hat{\mathbf{x}}_{s,k} = \hat{\mathbf{x}}_{f,k} + \mathbf{K}_{s,k}(\hat{\mathbf{x}}_{s,k+1} - \check{\mathbf{x}}_{f,k+1}), \quad (11)$$

$$\hat{\mathbf{P}}_{s,k} = \hat{\mathbf{P}}_{f,k} - \mathbf{K}_{s,k}(\check{\mathbf{P}}_{f,k+1} - \hat{\mathbf{P}}_{s,k+1})\mathbf{K}_{s,k}^\top, \quad (12)$$

are then used for $k = N-1, \dots, 0$, where $\mathbf{K}_{s,k}$ is the smoother gain, $\hat{\mathbf{x}}_{s,k}$ is the smoother estimate, and $\hat{\mathbf{P}}_{s,k}$ is the smoother covariance.

IV. MULTIPLICATIVE RTS SMOOTHING

The extended RTS smoother is used for nonlinear systems and has many variations, an example of which is discussed in [13]. Consider the nonlinear continuous-time process and discrete-time measurement models

$$\dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)), \quad (13)$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{X}(t), \mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)), \quad (14)$$

$$\mathbf{y}_k = \mathbf{g}_k(\mathbf{X}_k, \mathbf{x}_k, \mathbf{v}_k), \quad (15)$$

where $\mathbf{X} \in \mathcal{G}$ is an element of a matrix Lie group while $\mathbf{x} \in \mathbb{R}^{n_x}$, and both compose the state. For a process model of the form given by (13) and (14), the MRTS smoother can be used for state estimation. A similar multiplicative approach to smoothing is taken in [6], [15]. The MRTS smoother can be viewed as an extension of the extended RTS smoother in the same way the multiplicative extended Kalman Filter (MEKF) is an extension of the standard EKF. Owing to the popularity of the EKF, and the use of the MEKF as the benchmark when assessing the benefits of the IEKF, the MRTS smoother will be compared to the proposed IRTS smoother derived in Section V.

As is done in many Kalman filter variants, the MRTS smoother requires linearization of the process and measurement models. Combining and linearizing (13) and (14), and also linearizing (15), results in

$$\delta \dot{\mathbf{x}} = \mathbf{A} \delta \mathbf{x} + \mathbf{L} \delta \mathbf{w}, \quad (16)$$

$$\delta \mathbf{y}_k = \mathbf{H}_k \delta \mathbf{x}_k + \mathbf{M}_k \delta \mathbf{v}_k, \quad (17)$$

where $\delta \mathbf{w} = \bar{\mathbf{w}} + \mathbf{w}$ and $\delta \mathbf{v}_k = \bar{\mathbf{v}}_k + \mathbf{v}_k$, \mathbf{A} and \mathbf{L} are the process model Jacobians, and \mathbf{H}_k and \mathbf{M}_k are the measurement model Jacobians. The error definitions used to perform the linearization are [16]

$$\delta \mathbf{x} = \begin{bmatrix} \log(\mathbf{X}^{-1} \hat{\mathbf{X}})^\vee \\ \hat{\mathbf{x}} - \mathbf{x} \end{bmatrix}. \quad (18)$$

The error in the state defined on \mathcal{G} is multiplicative because \mathcal{G} is not closed under addition. Also, this is just one error definition; alternative definitions can be used as well.

The forward filter is the known MEKF [1], [15]. The prediction step is performed by integrating (13) and (14) from time t_{k-1} to time t_k using the expected value of the process noise, which is zero. The predicted covariance is given by (6) and the Kalman gain is given by (7). The state estimate correction is given by

$$\begin{bmatrix} \delta \chi_1 \\ \delta \chi_2 \end{bmatrix} = \mathbf{K}_{f,k} \mathbf{z}_k, \quad (19)$$

$$\hat{\mathbf{X}}_{f,k} = \check{\mathbf{X}}_{f,k} \exp(-\delta \chi_1^\wedge), \quad (20)$$

$$\hat{\mathbf{x}}_{f,k} = \check{\mathbf{x}}_{f,k} - \delta \chi_2, \quad (21)$$

where $\mathbf{z}_k = \mathbf{y}_k - \check{\mathbf{y}}_k$ and the corrected covariance is given by (9). Note that $\delta \chi_1^\wedge$ is an element of the Lie algebra associated with \mathcal{G} .

The backward smoothing is initialized using the state estimate and covariance output of the forward filter. The Kalman gain for the smoother is given by (10) and the smoother covariance update is given by (12). The smoother state innovation and smoother state update are given by

$$\begin{bmatrix} \mathbf{z}_{s,k}^1 \\ \mathbf{z}_{s,k}^2 \end{bmatrix} = \begin{bmatrix} \log(\hat{\mathbf{X}}_{s,k+1}^{-1} \check{\mathbf{X}}_{f,k+1})^\vee \\ \hat{\mathbf{x}}_{s,k+1} - \check{\mathbf{x}}_{f,k+1} \end{bmatrix}, \quad (22)$$

$$\hat{\mathbf{X}}_{s,k} = \hat{\mathbf{X}}_{f,k} \exp\left(-(\mathbf{K}_{s,k} \mathbf{z}_{s,k}^1)^\wedge\right), \quad (23)$$

$$\hat{\mathbf{x}}_{s,k} = \hat{\mathbf{x}}_{f,k} - \mathbf{K}_{s,k} \mathbf{z}_{s,k}^2. \quad (24)$$

V. INVARIANT RTS SMOOTHING

The invariant extended Kalman filter (IEKF) [3] can be considered a variant of the extended Kalman filter (EKF), but specific to estimation problems with particular properties. Here, the invariant filtering framework is leveraged to introduce the invariant RTS (IRTS) smoother. Similar to traditional RTS and MRTS smoothing, the forward filter in IRTS smoothing is the IEKF. Subsequently, a backward smoothing step is performed. The IEKF and proposed IRTS smoother are applicable to process models of the form

$$\dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t), \mathbf{u}(t)) + \mathbf{X}(t) \mathbf{W}(t), \quad (25)$$

where $\mathbf{X}(t) \in \mathcal{G}$ is the state and $\mathbf{W}(t) \in \mathfrak{g}$ is process noise where $\mathbf{w}(t) = \mathbf{W}(t)^\vee \in \mathbb{R}^m$. The multiplication between

the state and process noise in (25) is necessary since matrix Lie groups are closed under multiplication but not under addition. The argument (t) will once again be omitted for brevity. Given the true state \mathbf{X} and the state estimate $\hat{\mathbf{X}}$, the error can be defined as a left-invariant error or right-invariant error. Additionally, when the process model (excluding noise) is group affine, the error dynamics are state-independent, as stated in Theorem 1.

Theorem 1 (State-independent error dynamics [3]): If the function $\mathbf{F}(\mathbf{X}(t), \mathbf{u}(t))$ is group affine and the error is either left- or right-invariant, then the error propagation will be state independent.

The choice of using a left- or right-invariant error depends on the left- or right-invariant form of the measurements, as defined next.

Definition 3 (Left- and right-invariant measurement model): Left- and right-invariant measurements are

$$\mathbf{y}_k^L = \mathbf{X}_k \mathbf{b}_k + \mathbf{v}_k, \quad (26)$$

$$\mathbf{y}_k^R = \mathbf{X}_k^{-1} \mathbf{b}_k + \mathbf{v}_k, \quad (27)$$

respectively, where \mathbf{b}_k is some known column matrix of appropriate dimension.

When confronted with a left-invariant (right-invariant) measurement model and a group affine process model, a left-invariant error (right-invariant error) should be used [3], [4].

The error dynamics are linearized using $\delta \mathbf{X} = \exp(\delta \xi^\wedge) \approx \mathbf{1} + \delta \xi^\wedge$, where $\delta \xi \in \mathbb{R}^m$ is the state of the linearized system, and the superscript for left or right has been dropped here. The linearized process model is

$$\delta \dot{\xi} = \mathbf{A} \delta \xi + \mathbf{L} \delta \mathbf{w}. \quad (28)$$

The prediction step in the forward filter from \mathbf{X}_{k-1} to \mathbf{X}_k is performed by integrating (25) from time t_{k-1} to t_k . The predicted covariance is given by (6).

A. L-IRTS Smoothing

A detailed explanation of left-IRTS (L-IRTS) smoothing is provided here. Because L-IRTS smoothing and right-IRTS smoothing (R-IRTS) smoothing are required in the sample problem presented in Section VI, the equivalent equations for R-IRTS smoothing are provided in Section V-B.

The left-invariant error $\delta \mathbf{X}_{f,k} = \hat{\mathbf{X}}_{f,k}^{-1} \check{\mathbf{X}}_{f,k}$ is the error between the predicted and the corrected state. Rearranging the error definition results in an expression for the corrected state given by

$$\hat{\mathbf{X}}_{f,k} = \check{\mathbf{X}}_{f,k} \delta \mathbf{X}_{f,k}^{-1} = \check{\mathbf{X}}_{f,k} \exp\left(-(\mathbf{K}_{f,k} \mathbf{z}_{f,k}^L)^\wedge\right), \quad (29)$$

where the left-innovation is given by

$$\mathbf{z}_{f,k}^L = \check{\mathbf{X}}_{f,k}^{-1} (\mathbf{y}_k - \check{\mathbf{y}}_k), \quad (30)$$

and the Kalman gain is given by (7), $\check{\mathbf{y}}_k = \check{\mathbf{X}}_{f,k} \mathbf{b}_k$, and the negative sign in the exponent is a result of the inverse operation. To find the Jacobians \mathbf{H}_k and \mathbf{M}_k the innovation must be linearized. To do so, substitute (26) into the innovation expression and use the definition of the error, which results in

$$\delta \mathbf{z}_{f,k}^L = \mathbf{H}_k \delta \check{\xi}_{f,k} + \mathbf{M}_k \delta \mathbf{v}_k. \quad (31)$$

After the forward-filter pass, the backward smoothing is initialized with the corrected state estimate $\hat{\mathbf{X}}_{f,N}$ and covariance $\hat{\mathbf{P}}_{f,N}$. The Kalman gain used in the backward smoothing is given by (10) and the covariance update is given by (12). To find an expression describing the smoothed state estimate, the error between $\hat{\mathbf{X}}_{s,k+1}$ and $\check{\mathbf{X}}_{f,k+1}$ must be defined. In the traditional RTS smoother there is the $\hat{\mathbf{x}}_{s,k+1} - \check{\mathbf{x}}_{f,k+1}$ term which can be seen as a $\delta\mathbf{x}$ term or as a smoother innovation term. The equivalent error in the present context, when the state is an element of the matrix Lie group \mathcal{G} and a left-invariant error is employed, is

$$\delta\mathbf{X}_{s,k} = \hat{\mathbf{X}}_{s,k+1}^{-1} \check{\mathbf{X}}_{f,k+1}, \quad (32)$$

where $\exp(\delta\xi_{s,k}^\wedge) = \delta\mathbf{X}_{s,k}$. Using the definition of $\delta\xi_{s,k}$, an expression for the left smoother-state innovation $\mathbf{z}_{s,k}^L$ is

$$\mathbf{z}_{s,k}^L = \delta\xi_{s,k} = \log\left(\hat{\mathbf{X}}_{s,k+1}^{-1} \check{\mathbf{X}}_{f,k+1}\right)^\vee. \quad (33)$$

Given that the smoother-state innovation has been defined, an expression for the smoother state estimate update can now be found. The left-invariant IRTS smoother update is then

$$\hat{\mathbf{X}}_{s,k} = \hat{\mathbf{X}}_{f,k} \exp\left(-(\mathbf{K}_{s,k} \mathbf{z}_{s,k}^L)^\wedge\right). \quad (34)$$

B. R-IRTS Smoothing

The right-invariant error associated with the filter is

$$\delta\mathbf{X}_{f,k} = \check{\mathbf{X}}_{f,k} \hat{\mathbf{X}}_{f,k}^{-1}, \quad (35)$$

which justifies the form of the state correction,

$$\hat{\mathbf{X}}_{f,k} = \exp\left(-(\mathbf{K}_{f,k} \mathbf{z}_{f,k}^R)^\wedge\right) \check{\mathbf{X}}_{f,k}, \quad (36)$$

where the right-innovation is given by

$$\mathbf{z}_{f,k}^R = \check{\mathbf{X}}_k (\mathbf{y}_k - \check{\mathbf{y}}_k). \quad (37)$$

The innovation and state-estimate update used in the smoothing phase are

$$\mathbf{z}_{s,k}^R = \log\left(\check{\mathbf{X}}_{f,k+1} \hat{\mathbf{X}}_{s,k+1}^{-1}\right)^\vee, \quad (38)$$

$$\hat{\mathbf{X}}_{s,k} = \exp\left(-(\mathbf{K}_{s,k} \mathbf{z}_{s,k}^R)^\wedge\right) \check{\mathbf{X}}_{f,k}. \quad (39)$$

VI. SAMPLE PROBLEM: $SE(3)$ WITH BIAS

A. Problem Formulation

Consider a rigid body free to rotate and translate in three-dimensional space. Let \mathcal{F}_a be an inertial frame composed of three orthonormal physical basis vectors [17] and \mathcal{F}_b be a frame that rotates with the body. The orientation of \mathcal{F}_a relative to \mathcal{F}_b is described by a direction cosine matrix (DCM) $\mathbf{C}_{ab} \in SO(3)$. A physical vector \underline{v} can be resolved in \mathcal{F}_a as a column matrix \mathbf{v}_a or in \mathcal{F}_b as \mathbf{v}_b , where $\mathbf{v}_a = \mathbf{C}_{ab} \mathbf{v}_b$ and $\mathbf{v}_a, \mathbf{v}_b \in \mathbb{R}^3$. Point w is a datum point, and point z is fixed to the body. The position of point z relative to point w resolved in \mathcal{F}_a is denoted as $\mathbf{r}_a^{zw} \in \mathbb{R}^3$. The velocity of z relative to w with respect to \mathcal{F}_a is denoted as $\dot{\mathbf{r}}_a^{zw} = \mathbf{v}_a^{zw/a}$ and the angular velocity of \mathcal{F}_b relative to \mathcal{F}_a resolved in \mathcal{F}_b is given by $\boldsymbol{\omega}_b^{ba} \in \mathbb{R}^3$.

Consider noisy, biased, angular velocity measurements provided by a rate gyro of the form

$$\mathbf{u}_b^1 = \boldsymbol{\omega}_b^{ba} - \boldsymbol{\beta}_b^1 - \mathbf{w}_b^1, \quad (40)$$

where $\mathbf{w}_b^1 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^1)$, and $\boldsymbol{\beta}_b^1$ is the measurement bias. Additionally, consider noisy, biased, translational velocity measurements given by

$$\mathbf{u}_b^2 = \mathbf{C}_{ab}^\top \mathbf{v}_a^{zw/a} - \boldsymbol{\beta}_b^2 - \mathbf{w}_b^2, \quad (41)$$

where $\mathbf{w}_b^2 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^2)$, and $\boldsymbol{\beta}_b^2$ is sensor bias. Both biases are random walk processes where $\dot{\boldsymbol{\beta}}_b^1 = \mathbf{w}_b^3$, $\dot{\boldsymbol{\beta}}_b^2 = \mathbf{w}_b^4$, and $\mathbf{w}_b^3 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^3)$, $\mathbf{w}_b^4 \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}^4)$. The states of interest for estimation are the attitude \mathbf{C}_{ab} , position \mathbf{r}_a^{zw} , and sensor biases $\boldsymbol{\beta}_b^1$ and $\boldsymbol{\beta}_b^2$. Similarly to [18], these states can be cast into an element of a matrix Lie group \mathcal{G} as

$$\mathbf{X} = \begin{bmatrix} \mathbf{C}_{ab} & \mathbf{r}_a^{zw} & & & \\ & 1 & & & \\ & & \mathbf{1} & \boldsymbol{\beta}_b^1 & \boldsymbol{\beta}_b^2 \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \in \mathcal{G}. \quad (42)$$

Details pertaining to the matrix Lie group \mathcal{G} are presented in the Appendix.

The kinematics of the problem are given by

$$\dot{\mathbf{C}}_{ab} = \mathbf{C}_{ab} \boldsymbol{\omega}_b^{ba \times}, \quad \dot{\mathbf{r}}_a^{zw} = \mathbf{v}_a^{zw/a}, \quad (43)$$

where $(\cdot)^\times : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ is the linear operator that maps a three dimensional column matrix to the matrix Lie algebra $\mathfrak{so}(3)$. The continuous-time process model is then given by

$$\dot{\mathbf{C}}_{ab} = \mathbf{C}_{ab} (\mathbf{u}_b^1 + \boldsymbol{\beta}_b^1 + \mathbf{w}_b^1)^\times, \quad \dot{\boldsymbol{\beta}}_b^1 = \mathbf{w}_b^3, \quad (44)$$

$$\dot{\mathbf{r}}_a^{zw} = \mathbf{C}_{ab} (\mathbf{u}_b^2 + \boldsymbol{\beta}_b^2 + \mathbf{w}_b^2), \quad \dot{\boldsymbol{\beta}}_b^2 = \mathbf{w}_b^4. \quad (45)$$

The inclusion of sensor biases in the problem violates the group affine properties required by the invariant framework, meaning the kinematics do not satisfy (1). However, the use of the invariant error in the linearization leads to Jacobians that are less state-estimate dependent than Jacobians computed using a multiplicative error definition.

A position measurement resolved in \mathcal{F}_a , such as a global positioning system (GPS) measurement, provides

$$\mathbf{y}_{a,k} = \mathbf{r}_a^{z_k w} + \mathbf{v}_{a,k}, \quad (46)$$

where $\mathbf{v}_{a,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k^1)$, which can also be written as a function of the state \mathbf{X}_k ,

$$\begin{bmatrix} \mathbf{y}_{a,k} \\ 1 \\ \mathbf{0} \end{bmatrix} = \mathbf{X}_k \begin{bmatrix} \mathbf{0} \\ 1 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{a,k} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (47)$$

where the matrices $\mathbf{0}$ are of appropriate size. The position measurement is left-invariant because it is of the form given in (26). In addition, landmark measurements resolved in \mathcal{F}_b , from a LIDAR or stereo camera, for example, are available. Denote point p^i to be the i^{th} landmark. The position of the i^{th} landmark relative to w resolved in \mathcal{F}_a is given by $\mathbf{r}_a^{p_i w}$. The landmark sensor measures

$$\mathbf{y}_{b_k}^i = \mathbf{C}_{ab_k}^\top (\mathbf{r}_a^{p_i w} - \mathbf{r}_a^{z_k w}) + \mathbf{v}_{b_k}^i, \quad (48)$$

where $\mathbf{v}_{b_k}^2 \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k^2)$, which can alternatively be written

$$\begin{bmatrix} \mathbf{y}_{b_k}^2 \\ 1 \\ \mathbf{0} \end{bmatrix} = \mathbf{X}_k^{-1} \begin{bmatrix} \mathbf{r}_a^{p_a, w} \\ 1 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{b_k}^2 \\ 0 \\ \mathbf{0} \end{bmatrix} \quad (49)$$

The landmark sensor measurement is right-invariant because it is of the form given in (27).

B. Using both Left- and Right-Invariant Measurements

Generally, the choice of the left- or right-invariant error is dictated by the left- or right-invariant form of the exteroceptive measurements, as described in Section V. For instance, given only a left-invariant measurement, such as GPS, the L-IRTS smoothing approach of Section V-A would be employed. However, when both left- and right-invariant measurements are available, such as in this paper, in order to maintain innovation Jacobians in the forward pass that are state-estimate independent, an error representation transformation (ERT) between left- and right-invariant errors is done, as discussed in [19]. Nominally, a left-invariant correction, given by (29), and covariance computation, given by (9), are computed in the forward pass of the smoother. When a right-invariant measurement is available, the left-invariant covariance, $\check{\mathbf{P}}_k^L$, is mapped to the right-invariant covariance, $\check{\mathbf{P}}_k^R$, using the matrix representation of the adjoint operator,

$$\check{\mathbf{P}}_k^R = \text{Ad}(\check{\mathbf{X}}_k) \check{\mathbf{P}}_k^L \text{Ad}(\check{\mathbf{X}}_k)^\top. \quad (50)$$

The right-invariant measurement is then used within a right-invariant correction step, given by (36), and covariance correction, again using (9) but with the appropriate right-invariant Jacobians. The corrected covariance in right-invariant form is then mapped back to a covariance in left-invariant form using an analogous version of (50). This ERT of the covariance between left- and right-invariant forms ensures that the Jacobians used within the correction step and covariance computation (9) are state-estimate independent.

When solving the batch state estimation problem with both left- and right-invariant measurement using IGN the measurements are not processed sequentially. Rather, all measurements, both left- and right-invariant, are used simultaneously. Therefore, when using a left-invariant error and innovation, although the Jacobians associated with the left-invariant measurement will be state-estimate independent, the Jacobians associated with the right-invariant measurement will depend on the state estimate. As such, an advantage of the IRTS smoother over IGN is that the ERT allows for computation of Jacobians that are consistent with the left- or right-invariant form of the measurement.

C. Error Propagation

To derive the error propagation in the IRTS, a left-invariant error of the form $\delta \mathbf{X} = \mathbf{X}^{-1} \hat{\mathbf{X}}$ is chosen. For this particular problem, the use of a left-invariant error leads to process model Jacobians that are less state-estimate dependent than when a right-invariant error is used.

Linearizing the process model using the left-invariant error definition leads to

$$\delta \dot{\xi} = \mathbf{A} \delta \xi + \mathbf{L} \delta \mathbf{w}, \quad (51)$$

where

$$\mathbf{A} = \begin{bmatrix} -(\mathbf{u}_b^1 + \hat{\beta}_b^1)^\times & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ -(\mathbf{u}_b^2 + \hat{\beta}_b^2)^\times & -(\mathbf{u}_b^1 + \hat{\beta}_b^1)^\times & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{L} = -\mathbf{1}, \quad (52)$$

and $\delta \xi = [\delta \xi^{\phi^\top} \quad \delta \xi^{\mathbf{r}^\top} \quad \delta \xi^{\beta^1 \top} \quad \delta \xi^{\beta^2 \top}]^\top$. Notice that \mathbf{A} depends on the measurements \mathbf{u}_b^1 and \mathbf{u}_b^2 , the bias estimates $\hat{\beta}_b^1$ and $\hat{\beta}_b^2$, but not on any other state estimates.

The continuous-time process model can then be discretized using any desired discretization scheme. In this paper, a forward Euler discretization scheme is chosen for simplicity.

D. Linearization of the Measurement Model

For the left-invariant measurements the innovation term is given by (30), and for the right-invariant measurements the innovation term is given by (37). To find the Jacobians associated with the position measurements, the innovation $\mathbf{z}_{f,k}^L$ is linearized using the left-invariant error definition. Similarly, to find the Jacobians associated with the landmark measurements, the innovation $\mathbf{z}_{f,k}^R$ is linearized using the right-invariant error definition. Linearizing the innovation $\mathbf{z}_{f,k}^L$ leads to measurement model Jacobians of the form

$$\mathbf{H}_k^L = [\mathbf{0} \quad -\mathbf{1} \quad \mathbf{0} \quad \mathbf{0}], \quad \mathbf{M}_k^L = \check{\mathbf{C}}_{ab_k}^\top. \quad (53)$$

For each of the m landmarks, linearizing the innovation $\mathbf{z}_{f,k}^R$ using the right-invariant error definition leads to measurement model Jacobians of the form

$$\mathbf{H}_k^{R,i} = \text{row}_{i=1, \dots, m} \left[-\mathbf{r}_a^{p_a, w \times} \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \right], \quad (54)$$

$$\mathbf{M}_k^R = \text{diag}(\check{\mathbf{C}}_{ab_k}, \dots, \check{\mathbf{C}}_{ab_k}). \quad (55)$$

Note that the Jacobians \mathbf{H}_k^L and $\mathbf{H}_k^{R,i}$ are constant and do not depend on the state, as the position of the landmarks $\mathbf{r}_a^{p_a, w}$ are known.

E. Jacobians using MRTS smoothing

To implement the MRTS smoother, attitude error is taken to be multiplicative, while position and bias errors are taken to be additive. The error definitions for the MRTS smoother used in the linearization of the process and measurement model are given by

$$\delta \mathbf{C} = \mathbf{C}_{ab}^\top \hat{\mathbf{C}}_{ab}, \quad \delta \mathbf{r}_a^{zw} = \hat{\mathbf{r}}_a^{zw} - \mathbf{r}_a^{zw}, \quad (56)$$

$$\delta \beta_b^1 = \hat{\beta}_b^1 - \beta_b^1, \quad \delta \beta_b^2 = \hat{\beta}_b^2 - \beta_b^2. \quad (57)$$

The Jacobians associated with MRTS smoother, denoted by the $(\cdot)^*$, are given by

$$\mathbf{A}^* = \begin{bmatrix} -(\mathbf{u}_b^1 + \hat{\beta}_b^1)^\times & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ -\hat{\mathbf{C}}_{ab} (\mathbf{u}_b^2 + \hat{\beta}_b^2)^\times & \mathbf{0} & \mathbf{0} & \hat{\mathbf{C}}_{ab} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad (58)$$

$$\mathbf{L}^* = -\text{diag}(\mathbf{1}, \hat{\mathbf{C}}_{ab}, \mathbf{1}, \mathbf{1}). \quad (59)$$

The measurement model Jacobians for the MRTS smoother are derived by linearizing the innovation $\mathbf{z}_{f,k} = \mathbf{y}_k - \check{\mathbf{y}}_k$ using the error definitions given by (56). The position measurement Jacobians are given by

$$\mathbf{H}_k^{1*} = [\mathbf{0} \quad -\mathbf{1} \quad \mathbf{0} \quad \mathbf{0}], \quad \mathbf{M}_k^{1*} = \mathbf{1}. \quad (60)$$

The Jacobians corresponding to the landmark sensor measurements for the MRTS smoother are given by

$$\mathbf{H}_k^{2,i*} = \underset{i=1,\dots,m}{\text{row}} \left[-(\check{\mathbf{C}}_{abk} (\mathbf{r}_a^{p,w} - \check{\mathbf{r}}_a^{z,w}))^\times \quad \check{\mathbf{C}}_{abk}^\top \quad \mathbf{0} \quad \mathbf{0} \right], \quad (61)$$

$$\mathbf{M}_k^{2*} = \mathbf{1}. \quad (62)$$

Notice that the process model Jacobians derived using the error definitions for the MRTS smoother, (58) and (59), depend on the attitude estimate $\hat{\mathbf{C}}_{ab}$, and not just the bias estimates like the Jacobians used in the IRTS smoother in (52). In addition, the measurement model Jacobian $\mathbf{H}_k^{2,i*}$ given in (61) depends on the attitude estimate and the position estimate, while the Jacobian for the IRTS smoother, $\mathbf{H}_k^{\text{R},i}$ given in (54), is constant.

VII. EXPERIMENTAL RESULTS USING THE STARRY NIGHT DATASET

The MRTS and the IRTS smoothers were first tested using simulated data, demonstrating encouraging results. To test the smoothers on a real-life problem, the IRTS and MRTS smoothers were compared using the Starry Night dataset [10]. The Starry Night dataset provides angular and translational velocity sensor measurements at approximately 100 Hz, and stereo image pairs logged at approximately 15 Hz. The camera measurements are preprocessed as in [5] to obtain a right-invariant measurement model of the form (49). These preprocessed camera measurements are used in both the IRTS and MRTS smoothers. The angular and translational velocity sensor data provided in the dataset are unbiased. To add complexity to the experiments, large artificial biases are added to the angular and translational velocity sensor data. The initial biases are set to $\beta_{b,0}^1 = [0.05 \quad 0.05 \quad 0.05]^\top \text{ rad s}^{-1}$ and $\beta_{b,0}^2 = [0.04 \quad -0.03 \quad 0.06]^\top \text{ m s}^{-1}$. For all experiments, the standard deviation on the bias random walks are both set to $\sigma_{\beta_1} = \sigma_{\beta_2} = 0.005$ with appropriate units. In addition, artificial position measurements of the form (47) are generated using ground truth data at approximately 10 Hz. The same standard deviation of $\sigma_{R_1} = 0.5 \text{ m}$ is used for the artificial position measurements in all subsequent experiments.

A 20 second window of the Starry Night dataset is chosen to test the smoothers. The smoothers are tested for

both low and high initialization errors. Figure 1 shows the root mean square error (RMSE) in each state for 100 Monte-Carlo simulations using the IRTS and the MRTS smoothers with a low initialization error. The initial error for each Monte-Carlo simulation is sampled from the distribution $\delta\xi_0 \sim \mathcal{N}(\mathbf{m}, \mathbf{P})$, with the initial mean error set to $\mathbf{m} = [m_\phi \mathbf{1}_{1:3}^\top \quad m_r \mathbf{1}_{1:3}^\top \quad m_{\beta_1} \mathbf{1}_{1:3}^\top \quad m_{\beta_2} \mathbf{1}_{1:3}^\top]^\top$, where $\mathbf{1}_{1:3} = [1 \ 1 \ 1]^\top$, $m_\phi = \frac{\pi}{12} \text{ rad}$, $m_r = 0.1 \text{ m}$, $m_{\beta_1} = 0.005 \text{ rad s}^{-1}$, and $m_{\beta_2} = 0.005 \text{ m s}^{-1}$. The covariance on the initial error distribution is set to $\mathbf{P} = \text{diag}(\sigma_\phi^2 \mathbf{1}, \sigma_r^2 \mathbf{1}, \sigma_{\beta_1}^2 \mathbf{1}, \sigma_{\beta_2}^2 \mathbf{1})$, where $\sigma_\phi = \frac{\pi}{36} \text{ rad}$, $\sigma_r = 0.1 \text{ m}$, $\sigma_{\beta_1} = 0.005 \text{ rad s}^{-1}$, and $\sigma_{\beta_2} = 0.005 \text{ m s}^{-1}$. In addition, for each Monte-Carlo trial, new realizations of the artificial biases and artificial position measurements are generated. The initial state estimate for the IRTS smoother is generated using the left-invariant error definition, while the initial state estimate for the MRTS smoother is generated using the multiplicative error definitions given by (56). The lower bounds of the error bars in Figure 1, and all forthcoming figures, are set to a percentile of 2.5 and the upper bounds of the error bars are set to a percentile of 97.5, meaning the results of 95% of the trials lie within the error bars.

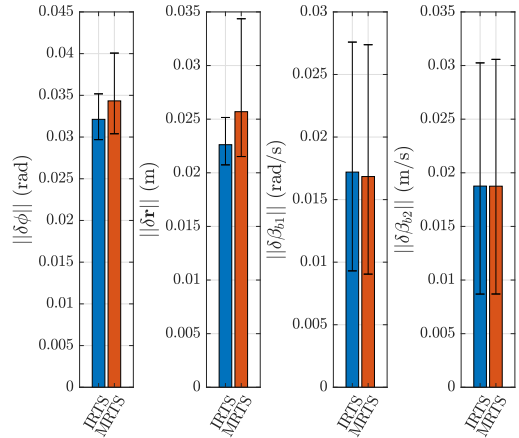


Fig. 1: Mean RMSEs for the IRTS and MRTS smoothers on experimental data with low initialization error.

With a low initialization error, Figure 1 indicates that the performance of the IRTS and MRTS smoothers are comparable, with the IRTS smoother performing slightly better on average. The advantage of the IRTS smoother lies in the case of poor initialization, as the Jacobians of the IRTS smoother are less state-estimate dependent and are therefore more accurate even when the state estimate is poor. To demonstrate this, both filters were tested for a large initialization error in all states. The mean initial errors for each state are set to $m_\phi = \frac{\pi}{3} \text{ rad}$, $m_r = 1 \text{ m}$, $m_{\beta_1} = 0.03 \text{ rad s}^{-1}$, and $m_{\beta_2} = 0.03 \text{ m s}^{-1}$. Figure 2 presents the results of 100 Monte-Carlo simulations. Figure 2 demonstrates that when the smoothers are poorly initialized, the IRTS smoother outperforms the MRTS smoother by a large margin.

The IRTS and MRTS smoothers are batch state estimators, which in turn motivates a comparison to IGN and MGN, for

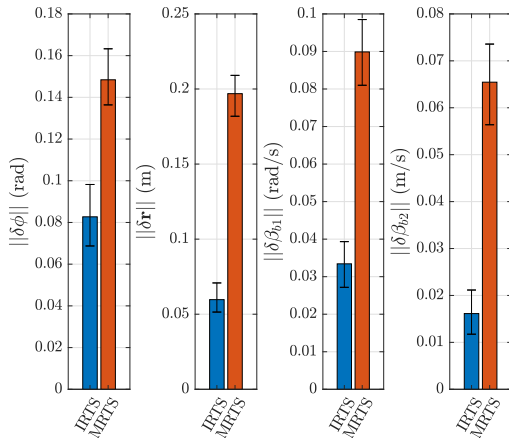


Fig. 2: Mean RMSEs for the IRTS and MRTS smoothers on experimental data with high initialization error.

one and more iterations. The nonlinear least squares problem associated with a *maximum a posteriori* (MAP) formulation of the state estimation problem is solved via GN optimization [2]. Within the IGN framework, a left-invariant error and innovation are employed with left-invariant measurements, while a left-invariant error and standard innovation is used with right-invariant measurements. Within the MGN framework, the multiplicative errors defined in (56) and a standard innovation are used for all measurements.

Next, a comparison of the IRTS and MRTS smoothers to both IGN and MGN is provided. One iteration of either the IRTS or the MRTS smoothers is defined as one forward pass followed by one backwards pass of the smoothing algorithm. One iteration of a smoothing algorithm is almost, but not exactly, equivalent to one iteration of GN [20]. For every subsequent iteration of the smoothers, the forward filter is then re-initialized with the new initial state and covariance estimates from the previous backwards pass. Both IGN and MGN are initialized using dead reckoning. The mean RMSEs after each iteration of the smoothers and GN estimators are compared over 100 Monte-Carlo trials in Figure 3. In each Monte-Carlo simulation, a high initialization error is tested with the same parameters used previously.

The results in Figure 3 demonstrate that one iteration of either smoothers far outperforms one iteration of either of the GN algorithms. The reason is that the Jacobians in the smoothers are computed using the best available state estimate in both the forward and backwards passes, meaning the Jacobians used in the smoothers are more accurate as the forward and backward passes are executed step by step. In both IGN and MGN, all Jacobians are evaluated using the state estimate from the previous least-squares solution. The Jacobians in the first iteration of each GN algorithm are inaccurate due to their computation using dead-reckoning starting from a large initial error. Even after 5 iterations of both IGN and MGN, there is still large variability in the mean RMSEs, as shown by the top of the error bars in Figure 3. Both of the smoothers show significantly smaller mean RMSEs after one iteration, but also much lower variability in the results. It

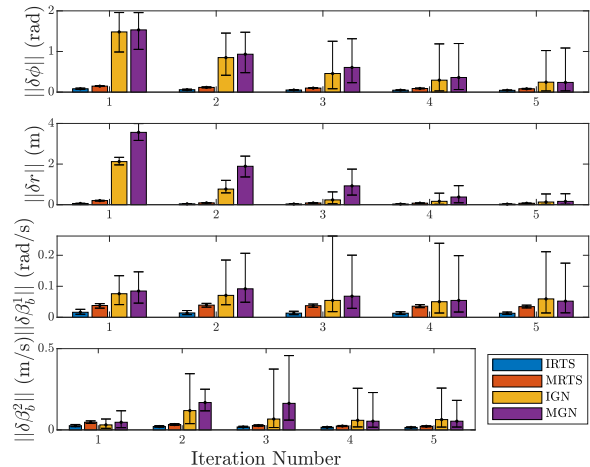


Fig. 3: Mean RMSEs for each smoother and Gauss-Newton algorithms for each iteration.

was noted that using a lower initialization error leads to a less drastic difference between one iteration of the smoothers and one iteration of GN approaches because the Jacobians used in the first iteration of the GN approaches are more accurate, and the solution converges faster.

There are further notable advantages to the IRTS smoother over IGN. For example, the covariance computation is straightforward in the IRTS smoother. In the forward pass, the covariance is computed using (9), and in the backward pass, the covariance is computed using (12). On the other hand, extracting the covariance associated with each state at each time step is cumbersome in IGN and MGN because a large, sparse, matrix must be inverted. Additionally, in a problem with both left- and right-invariant measurements, such as the problem presented in this paper, the ERT can be used, as discussed in Section VI-B. This leads to a correction step in the forward pass that is always consistent with the measurement type (i.e., consistent left- or right-invariant measurements). In an IGN framework, the innovation is fixed and cannot be changed, ultimately leading to Jacobians in the IGN framework that are state-estimate dependent in a problem with both left- and right-invariant measurements.

VIII. CONCLUSIONS

The main purpose of this paper is to present the IRTS smoother and benchmark it using experimental data relative to the MRTS smoother, IGN, and MGN. The IRTS smoother is essentially an application of the invariant filtering of [3], [4] where left- or right-invariant error definitions, group-affine process models, and left- or right-invariant measurement models are leveraged to give state-independent Jacobians. The IRTS and MRTS smoother were compared on an $SE(3)$ problem where the attitude and position of a body, and sensor biases, are estimated using angular velocity data, translational velocity data, stereo camera data, and GPS measurements simulated from ground-truth data. The IRTS smoother performance is quite good, performing as well or better than all

other algorithms. Specifically, when the initialization is poor, the IRTS smoother provides significantly better performance than the MRTS smoother. Furthermore, several iterations of IGN and MGN were required to match the performance of the IRTS smoother.

APPENDIX A SE(3) WITH BIAS

Three dimensional pose and two sensor biases can be cast into a matrix Lie group \mathcal{G} of the form

$$\mathbf{X} = \begin{bmatrix} \mathbf{C} & \mathbf{r} & & & & \\ & 1 & & & & \\ & & \mathbf{1} & \beta_b^1 & \beta_b^2 & \\ & & & 1 & & \\ & & & & & 1 \end{bmatrix} \in \mathcal{G}, \quad (63)$$

where $\mathcal{G} \in \mathbb{R}^{9 \times 9}$, $\mathbf{C} \in SO(3)$, and $\mathbf{r}, \beta_b^1, \beta_b^2 \in \mathbb{R}^3$, and zero entries have been omitted.

The matrix Lie algebra associated with \mathcal{G} is

$$\mathfrak{g} = \{ \Xi = \xi^\wedge \in \mathbb{R}^{12 \times 12} \mid \xi \in \mathbb{R}^{12} \}, \quad (64)$$

where

$$\xi^\wedge = \begin{bmatrix} \xi^{\phi^\times} \\ \xi^r \\ \xi^{\beta^1} \\ \xi^{\beta^2} \end{bmatrix} = \begin{bmatrix} \xi^{\phi^\times} & \xi^r & & & & \\ & 0 & & & & \\ & & \mathbf{0} & \xi^{\beta^1} & \xi^{\beta^2} & \\ & & & 0 & & \\ & & & & & 0 \end{bmatrix}. \quad (65)$$

Additionally, $\mathfrak{so}(3) = \{ \Omega = \omega^\times \in \mathbb{R}^{3 \times 3} \mid \omega \in \mathbb{R}^3 \}$ where

$$\omega^\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

The exponential map from \mathfrak{g} to \mathcal{G} is

$$\exp(\xi^\wedge) = \begin{bmatrix} \exp_{SO(3)}(\xi^{\phi^\times}) & \mathbf{J}\xi^r & & & & \\ & 1 & & & & \\ & & \mathbf{1} & \xi^{\beta^1} & \xi^{\beta^2} & \\ & & & 1 & & \\ & & & & & 1 \end{bmatrix} \in \mathcal{G}. \quad (66)$$

where $\exp_{SO(3)}(\xi^{\phi^\times})$ is given by Rodrigues' rotation formula [2] and \mathbf{J} is given by

$$\mathbf{J} = \frac{\sin \phi}{\phi} \mathbf{1} + \left(1 - \frac{\sin \phi}{\phi}\right) \mathbf{a}\mathbf{a}^\top + \frac{1 - \cos \phi}{\phi} \mathbf{a}^\times,$$

where $\phi = \|\xi^\phi\|$ and $\mathbf{a} = \xi^\phi / \phi$. The logarithmic map from \mathcal{G} to \mathfrak{g} is given by

$$\log(\mathbf{X}) = \begin{bmatrix} \log_{SO(3)}(\mathbf{C}) & \mathbf{J}^{-1}\mathbf{r} & & & & \\ & 0 & & & & \\ & & 0 & \beta_b^1 & \beta_b^2 & \\ & & & 0 & & \\ & & & & & 0 \end{bmatrix} \in \mathfrak{g}. \quad (67)$$

The adjoint operator for the group \mathcal{G} is given by

$$\text{Ad}(\mathbf{X}) = \begin{bmatrix} \mathbf{C} & & & & & \\ \mathbf{r}^\times \mathbf{C} & \mathbf{C} & & & & \\ & & \mathbf{1} & & & \\ & & & \mathbf{1} & & \\ & & & & \mathbf{1} & \\ & & & & & \mathbf{1} \end{bmatrix}. \quad (68)$$

ACKNOWLEDGMENT

The authors graciously acknowledge funding from Group for Research in Decision Analysis (GERAD) and the National Science and Engineering Research Council (NSERC) of Canada.

REFERENCES

- [1] E. Lefferts, L. Markley, and M. Shuster, "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, 1982.
- [2] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.
- [3] A. Barrau and S. Bonnabel, "The Invariant Extended Kalman Filter as a Stable Observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2016.
- [4] —, "Invariant Kalman Filtering," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 237–257, 2018.
- [5] J. Arsenaault, "Practical Considerations and Extensions of the Invariant Extended Kalman Filtering Framework," Master's thesis, McGill University, 2019.
- [6] G. Bourmaud, "Parameter Estimation on Lie Groups : Application to Mapping and Localization from a Monocular Camera," PhD. Thesis, Université de Bordeaux, Nov. 2015.
- [7] P. Chauchat, A. Barrau, and S. Bonnabel, "Invariant Smoothing on Lie Groups," *International Conference on Intelligent Robots and Systems (IROS)*, pp. 1703–1710, 2018.
- [8] P. Chauchat, "Smoothing Algorithms for Navigation, Localisation and Mapping Based on High-grade Inertial Sensors," Ph.D. dissertation, MINES ParisTech, 2020.
- [9] A. Walsh, J. Arsenaault, and J. R. Forbes, "Invariant Sliding Window Filtering for Attitude and Bias Estimation," *American Control Conference (ACC)*, pp. 3161–3166, 2019.
- [10] T. D. Barfoot, "AER1513 Course Notes, Assignments, and Data Sets," *University of Toronto, Institute for Aerospace Studies*, 2011.
- [11] B. C. Hall, *Lie Groups, Lie Algebras, and Representations*. 2nd ed. Springer, 2014.
- [12] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill Education, 2008.
- [13] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- [14] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. New York, NY, USA: Wiley-Interscience, 2006.
- [15] M. L. Psiaki, "Backward-Smoothing Extended Kalman Filter," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 5, pp. 885–894, 2005.
- [16] R. Aucoin, S. A. Chee, and J. R. Forbes, "Linear- and Linear-Matrix-Inequality-Constrained State Estimation for Nonlinear Systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 3153–3167, 2019.
- [17] P. Hughes, *Spacecraft Attitude Dynamics*. Wiley, 1986.
- [18] S. Heo and C. G. Park, "Consistent EKF-Based Visual-Inertial Odometry on Matrix Lie Group," *IEEE Sensors Journal*, vol. 18, no. 9, pp. 3780–3788, 2018.
- [19] M. R. Hartley, "Contact-Aided State Estimation on Lie Groups for Legged Robot Mapping and Control," Ph.D. dissertation, University of Michigan, 2019.
- [20] A. Y. Aravkin, J. V. Burke, and G. Pilonetto, *Optimization Viewpoint on Kalman Smoothing with Applications to Robust and Sparse Estimation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 237–280.