Characterizing the Uncertainty of Jointly Distributed Poses in the Lie Algebra

Joshua G. Mangelson, Maani Ghaffari, Ram Vasudevan, and Ryan M. Eustice

***Abstract*—An accurate characterization of pose uncertainty is essential for safe autonomous navigation. Early pose uncertainty characterization methods proposed by Smith, Self, and Cheeseman (SCC), used coordinate-based first-order methods to propagate uncertainty through non-linear functions such as pose composition (head-to-tail), pose inversion, and relative pose extraction (tail-to-tail). Characterizing uncertainty in the Lie Algebra of the special Euclidean group results in better uncertainty estimates. However, existing Lie group based uncertainty propagation techniques assume that individual poses are independent. After solving a pose graph, however, the entire trajectory is jointly distributed as factors induce correlation. Hence, the independence assumption does not capture reality. In addition, prior work has focused primarily on the pose composition operation. This paper develops a framework for modeling the uncertainty of jointly distributed poses and describes how to perform the equivalent of the SSC pose operations while characterizing uncertainty in the Lie Algebra. Evaluation on simulated and open-source datasets shows that the proposed methods result in more accurate uncertainty estimates and thus more accurate filtering of potential loop-closures. An accompanying C++ library implementation is also released.**

***Index Terms*—SLAM, mobile robotics, uncertainty propagation, Lie group, Lie algebra, matrix groups, rigid body transformation, state estimation.**

I. Introduction

An accurate characterization of robot pose (location and orientation) uncertainty is essential to robust long-term autonomy because planning and safety decisions are often predicated on their value [3]. For example, an over-confident position estimate could potentially result in a self-driving car crossing out of its lane or an underwater vehicle colliding with a submerged structure. On the other hand, under-confidence can lead to slow or sluggish behavior.

In addition to accurately characterizing the uncertainty of a single pose, it is often necessary to propagate the uncertainty of one or more poses through non-linear operations such as pose composition, pose inversion, and relative pose extraction. For example, given an odometry sequence consisting of multiple uncertain relative pose transformations, it may be necessary to compose the entire sequence of transformations together to obtain a distribution describing the relative pose of the robot over an entire sequence. Alternatively, after estimating the pose of the robot at multiple time steps via maximum likelihood estimation (MLE) or simultaneous localization and mapping (SLAM), it can be ueful to extract the relative pose of the robot between two time steps to evaluate whether a potential loop-closure is consistent with

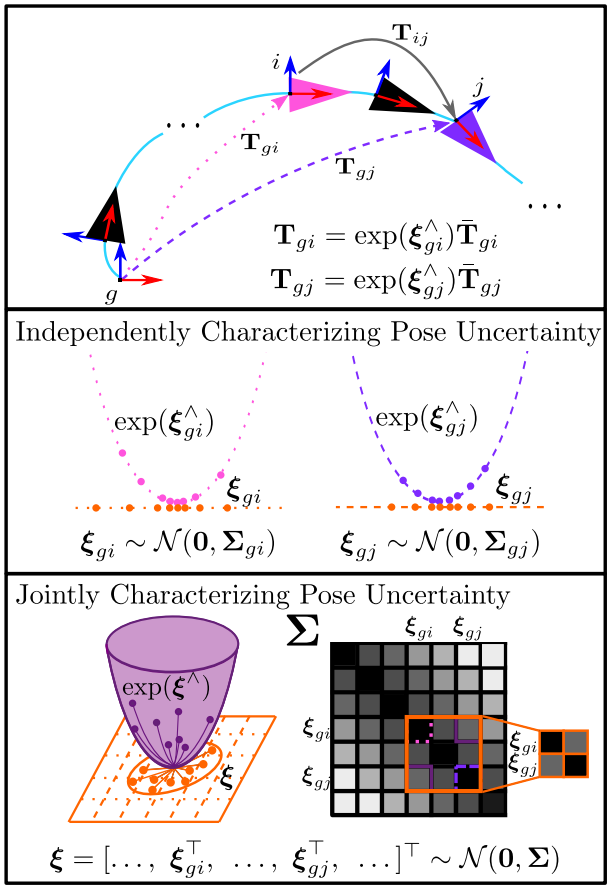


Fig. 1: State-of-the-art Pose-Graph SLAM algorithms estimate the pose (depicted as i and j in the top illustration) of a robotic vehicle at each time step with respect to a fixed coordinate frame, g, which is denoted by $\mathbf{T}_{gi}$ and $\mathbf{T}_{gj}$, respectively. After solving SLAM, it is often necessary to extract additional information by performing a variety of operations such as pose composition, pose inversion, and relative pose estimation, while accurately propagating uncertainty. An example of the relative pose operation $\mathbf{T}_{ij}$ is shown at the top of this figure. Recent work has shown that characterizing uncertainty as Gaussian random variables $(\boldsymbol{\xi}_{gi}, \boldsymbol{\xi}_{gj})$ in the Lie algebra of the Special Euclidean group (shown in the middle of the above figure) leads to increased consistency [1]; however, this approach has focused on pose composition while assuming that the underlying poses are independent. Typically, the poses estimated from SLAM are heavily correlated [2]. This paper proposes a framework for jointly characterizing the uncertainty of a set of correlated poses in the Lie algebra space (shown in the bottom illustration of the above figure). It then describes how to perform the pose composition, pose inverse, and relative pose operations within this framework.

*This work was supported by the Office of Naval Research under awards N00014-16-1-2102. Funding for M. Ghaffari is given by the Toyota Research Institute (TRI), partly under award number N021515, however this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

J. Mangelson, M. Ghaffari, R. Vasudevan, and R. Eustice are at the University of Michigan, Ann Arbor, MI 48109, USA. {mangelso, maanigj, ramv, eustice}@umich.edu.

the estimated trajectory.

A variety of methods for characterizing pose uncertainty have been proposed over the years. In particular, Smith, Self, and Cheeseman [4, 5] were some of the first to describe how to propagate pose uncertainty through the common pose operations mentioned above. Their proposed method represented the relative pose of objects using a multivariate Gaussian parameter vector (consisting of translation offsets and Euler angles) and used first order linearization techniques for uncertainty propagation. More recently, evaluations have shown that taking advantage of Lie group structure results in significantly more accurate pose uncertainty estimates [1, 6–8]. However, existing methods focus primarily on the pose composition operation, leaving out the pose inversion and relative pose operations.

In addition, existing Lie-based methods tend to assume that the poses being operated on are independent [1], an assumption that often does not hold true. For example, it is well known that the estimated poses obtained via a MLE algorithm such as Pose-Graph SLAM, tend to be heavily correlated [2]. Alternatively, wheel slip due to an irregular wheel or loose gravel may lead to sequence of short term odometry measurements being correlated over time. Ignoring this correlation when operating on jointly distributed poses results in inconsistency and over/under-confidence in the final uncertainty estimates, which in-turn can result in incorrect decision making and unsafe behaviour.

In this paper, we build on the work of Barfoot and Furgale [1] and describe how the uncertainty of multiple jointly distributed poses can be represented in the Lie algebra of the Special Euclidean group. However, we drop the independence assumption and additionally derive uncertainty propagation through the pose inversion and relative pose extraction operations (See Fig. 1). Finally, we show via both simulated and real-world experiments that the proposed method better characterizes uncertainty leading to better loop-closure filtering for SLAM [9, 10].

More specifically, the main contributions of this paper are the following:

1) we present a framework that describes how to represent jointly correlated poses while using the Lie algebra to characterize uncertainty;
2) we derive the equivalent of the SSC operations under the proposed framework;
3) we describe how to convert from alternative uncertainty characterization parameterizations to the proposed framework (including Lie algebra covariance extraction from a MLE solution);
4) we release an accompanying C++ library implementation along with examples presented here; and,
5) we evaluate the proposed method via experiments using both simulated and real-world data.

The remainder of this paper is organized as follows: Section II provides an overview of related work. Section III provides a brief introduction to the Special Euclidean group and some necessary concepts from Lie group theory. Section IV provides a summary of the SSC uncertainty representation framework and its associated operations. Section V describes how to use the Lie algebra to characterize uncertainty for jointly distributed poses. Sections VI, VII, and VIII describe the derivation of the pose composition, pose inversion, and relative pose operations, respectively, while characterizing uncertainty on the Lie algebra. Section IX describes how to convert from a coordinate based representation of uncertainty to the Lie algebra based representation and how to extract an estimate of pose uncertainty from a MLE solution. Section X describes an experimental evaluation of the proposed methods. Section XI contains some concluding thoughts, and finally the Appendix contains details of our open source C++ library implementation.

II. Related Work

Robotic perception of and interaction with real-world environments is inherently uncertain. Sensor measurements are corrupted by measurement noise and the motions invoked by actuators/motors are at-least partially un-predictable. Accurately and efficiently characterizing uncertainty is a fundamental requirement for reliable autonomous operation [3]. In particular, an accurate characterization of the relative pose of different coordinate frames is essential for navigation and interaction, since an inconsistent estimate of robot pose can result in collision or unsafe actions.

Various methods have been proposed to characterize the uncertainty of relative pose transformations. One of the earliest methods was the foundational work by Smith, Self, and Cheeseman [5] that used a vector of Euler angles and translation offsets to represent the pose of one or multiple coordinate frames and characterized uncertainty by modeling that vector of parameters as being derived from a multivariate Gaussian distribution. In addition to describing a method for characterizing pose uncertainty, Smith, Self, and Cheeseman described how first-order linearization could be used to propagate uncertainty through common operations such as pose composition (head-to-tail), pose inversion, and relative pose extraction(tail-to-tail). For brevity, the first-order jacobian uncertainty propagation based operations proposed in [5] are often referred to by the initials of the paper's authors (SSC). One of the nice properties of this method is that modeling the pose parameter vector as a multivariate Gaussian can easily be extended to model multiple poses as jointly Gaussian, enabling the ability to account for correlation between poses. However, first-order local linearization is heavily dependent on linearization point and ignores all non-linear aspects of the operation being performed.

Other methods try to improve approximation accuracy by incorporating non-linear information into the approximation process. The unscented transform (UT) [11] propagates a set of deterministic sample points through the non-linear operation and then uses a weighted average to approximate the final distribution as a Gaussian. Laplace approximation [12] follows a similar derivation to first-order approximation, but also takes into account second order derivative information. These methods result in more accurate Gaussian approximations than first-order methods when the true resulting distribution is heavily non-Gaussian at the expense of greater computational cost.

It has also been recently shown that the choice of pose parameterization has a significant effect on the accuracy and consistency of uncertainty propagation [8, 13]. Modeling a local coordinate pose representation, such as the parameter vector of Euler angles proposed in [5] as a multi-variate Gaussian can not possibly capture the true distribution since an uncertain pose is not actually Gaussian within that paramater space [7]. Using this choice of parameterization can lead to inconsistencies in estimation [14] and the loss of monotonicity in uncertainty propagation [13]. However, this can be overcome by using Lie group theory to account for the structure of the three dimensional (3D) Special Euclidean group [15–17], $\mathrm{SE}(3)$, since in the Lie algebra noise can be represented accurately as a multivariate Gaussian. Some of the first publications to do this were Chirikjian [18], Chirikjian [19], Wang and Chirikjian [6], and Long et al. [7] who represented an uncertain pose via *exponential coordinates* located in the Lie algebra of $\mathrm{SE}(3)$. Modeling pose uncertainty in this way overcomes the fact that in local coordinates pose uncertainty is inherently non-Gaussian and enables the representation of the so called banana distribution over robot position.

Barfoot and Furgale [1] were then able to show that propagation computations could be simplified by modeling the uncertainty directly in the Lie algebra and then using the exponential map to induce a distribution in the group space. Since the Lie algebra is a vector space, a small perturbation term can be modeled as zero-mean Gaussian noise in $\mathbb{R}^6$ and then used to perturb a mean (or nominal) pose in the group space. Barfoot and Furgale [1] then build on this by deriving first and second order uncertainty propagation for the pose composition operation when the associated poses are independent.

There have also been a variety of methods proposed for extracting estimates of uncertainty from the solutions to estimation algorithms. Kalman filter-based methods such as EKF-SLAM [20–23] explicitly track the covariance matrix of the values they estimate, although the parameterization of that covariance is somewhat dependent on the specific algorithm or implementation [24]. It is also possible to extract estimates of uncertainty from maximum likelihood estimation (MLE)-based estimation methods. Kaess and Dellaert [25] propose a method that is able to extract a Gaussian uncertainty approximation for the estimated variables from the information matrix used to find the MLE Pose-Graph SLAM solution. Google Ceres [26] also has the capability to extract covariance matrices using similar methods to those proposed in [25]. While the extraction of pose uncertainty from existing estimation solutions is not the primary contribution of this article, it is important to point out that when multiple poses are estimated jointly whether via Kalman filter based methods or MLE, the resulting poses are often highly correlated and thus propagation methods that operate on those estimated poses need to be able to handle such correlation.

Our approach for modeling the uncertainty of a set of poses is similar to [1], however we drop the independence requirement since it is often not upheld in practice and additionally describe the pose inversion and relative pose extraction operations (See Fig. 3).

III. The Special Euclidean Group and Lie Group Theory

Estimation of the relative pose (position and orientation) between objects or coordinate frames in space is a common problem in robotic navigation, perception, and manipulation. Formally, we represent 3D relative pose transformations as elements of the *Special Euclidean group*. This section provides a brief introduction to the *Special Euclidean group* and relevant aspects of Lie group theory that are important during the subsequent derivation and discussion.

A. The Special Euclidean group

The *Special Euclidean group*, or $\mathrm{SE}(d)$, represents the space of homogeneous transformation matrices or the space of matrices that apply a rigid body rotation and translation to points in $\mathbb{R}^d$ (represented in homogeneous form). Formally, in three dimensions, $\mathrm{SE}(3)$ is defined as follows:

$$\mathrm{SE}(3) := \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \middle| \mathbf{R} \in \mathrm{SO}(3), \mathbf{t} \in \mathbb{R}^3 \right\}, \tag{1}$$

where $\mathrm{SO}(3)$ is the *Special Orthogonal group* is the space of valid rotation matrices:

$$\mathrm{SO}(3) := \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} | \mathbf{R}\mathbf{R}^\top = \mathbf{I}_3, \det \mathbf{R} = 1 \right\}, \tag{2}$$

and $\mathbf{I}_d$ is the identity matrix of dimension d. A variety of methods have been developed for parameterizing these objects such as Euler angles or quaternions [15].

Both $\mathrm{SE}(3)$ and $\mathrm{SO}(3)$ are matrix *Lie groups* meaning they are smooth manifolds that also satisfy the formal definition of a mathematical group [27, 28] with the standard matrix multiplication operation. Intuitively, this means that while the general group is non-linear, each of these groups can be locally approximated using a Euclidean vector space. Additionally, for any given point on the manifold, consider the set of all paths on the manifold that pass through that point. The set of all velocities (both in terms of direction and speed) of those paths at the given point form a vector space called the tangent space. The tangent space centered at the identity is called the *Lie algebra*. This relationship is depicted in Fig. 2.

The Lie algebra of $\mathrm{SO}(3)$ is denoted $\mathfrak{so}(3)$ and is the space of skew-symmetric 3×3 matrices [19, 27]:

$$\mathfrak{so}(3) := \left\{ \boldsymbol{\omega} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{\omega}^\top = -\boldsymbol{\omega} \right\} \tag{3}$$

This space is isomorphic to $\mathbb{R}^3$ since skew-symmetric matrices have zeros on the diagonal entries and the lower entries are completely identified by three upper entries (hence $\dim \mathfrak{so}(3) = 3$). Therefore, it is very convenient to work with $\mathbb{R}^3$ instead. Note, we can use the $\wedge$ operator to take an element of $\mathbb{R}^3$ and transform it to an element of $\mathfrak{so}(3)$:

$$\boldsymbol{\phi}^\wedge := \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathfrak{so}(3), \tag{4}$$

where $\boldsymbol{\phi} \in \mathbb{R}^3$. The $\vee$ operator denotes the inverse of $\wedge$.

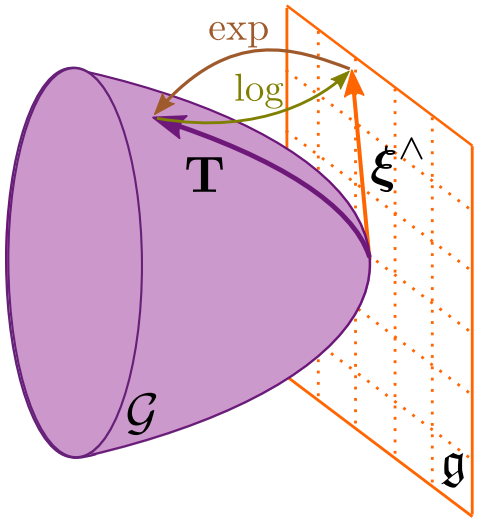


Fig. 2: The Lie algebra, $\mathfrak{g}$, is the tangent space to the Lie group, $\mathcal{G}$, centered at the identity element. The Lie algebra represents the space of all possible velocities a particle at a given point on the group could take. The *exponential map* maps velocities in the Lie algebra, $\boldsymbol{\xi}^\wedge \in \mathfrak{g}$, to their associated action in the Lie group, $\mathbf{T} \in \mathcal{G}$, and the *logarithm map* performs the inverse operation.

Similarly, the Lie algebra of $\mathrm{SE}(3)$, or $\mathfrak{se}(3)$, is defined as follows:

$$\mathfrak{se}(3) := \left\{ \begin{bmatrix} \boldsymbol{\omega} & \boldsymbol{\rho} \\ \mathbf{0}^\top & 0 \end{bmatrix} \;\middle|\; \boldsymbol{\omega} \in \mathfrak{so}(3), \boldsymbol{\rho} \in \mathbb{R}^3 \right\}. \tag{5}$$

This space of matrices is isomorphic to $\mathbb{R}^6$ and we overload the $\wedge$ operator to convert between the Euclidean vector and matrix forms:

$$\boldsymbol{\xi}^\wedge := \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^\wedge = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathfrak{se}(3), \tag{6}$$

where $\boldsymbol{\xi} \in \mathbb{R}^6$ and $\boldsymbol{\rho}, \boldsymbol{\phi} \in \mathbb{R}^3$.

Understanding the relationship between the group and algebra spaces can enable one to leverage the fact that the algebra is a vector space. The next few subsections cover some important concepts from Lie group Theory that we need in the rest of the paper. In doing so, we use $\mathcal{G}$ to represent a given Lie group and $\mathfrak{g}$ to represent its associated Lie algebra.

B. The Exponential Map

The Lie algebra, $\mathfrak{g}$, represents the tangent space of the manifold at the identity. However, given a specific tangent vector, we may want to convert it to its associated transformation in the group space $\mathcal{G}$. The *exponential map*, $\exp : \mathfrak{g} \to \mathcal{G}$, (which can be defined in closed form for $\mathrm{SE}(3)$), enables us to perform this conversion,

$$\exp(\boldsymbol{\xi}^\wedge) = \sum_{k=0}^{\infty} \frac{(\boldsymbol{\xi}^\wedge)^k}{k!} = \mathbf{I}_4 + \boldsymbol{\xi}^\wedge + \frac{(\boldsymbol{\xi}^\wedge)^2}{2} + \cdots. \tag{7}$$

The *logarithm map*, $\log : \mathcal{G} \to \mathfrak{g}$, on the other hand, enables us to go the other direction from an action/transformation in the group space to the velocity that would have induced it,

$$\log(\mathbf{T}) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(\mathbf{T} - \mathbf{I}_4)^k}{k}. \tag{8}$$

This relationship is visualized in Fig. 2.

C. The Adjoint Action

Assuming $\mathbf{T} \in \mathcal{G}$ and $\boldsymbol{\xi} \in \mathfrak{g}$, the adjoint action of $\mathbf{T}$ on $\boldsymbol{\xi}$, or $\mathrm{Ad}_{\mathbf{T}}(\boldsymbol{\xi})$ is defined as follows:

$$\mathrm{Ad}_{\mathbf{T}}(\boldsymbol{\xi}) := \mathrm{Ad}_{\mathbf{T}}\,\boldsymbol{\xi} = \log(\mathbf{T}\exp(\boldsymbol{\xi})\mathbf{T}^{-1}). \tag{9}$$

The adjoint action describes the affect that transforming to the group space applying a transformation on the left and its inverse on the right has on an element of the Lie algebra. This gives rise to the following property, which we use later on:

$$\mathbf{T}\exp(\boldsymbol{\xi}) = \exp(\mathrm{Ad}_{\mathbf{T}}\,\boldsymbol{\xi})\mathbf{T} \tag{10}$$

D. The Baker-Campbell-Hausdorff (BCH) Formula

Finally, we also want to characterize the effect that multiplication in the group space has on the Lie algebra. More specifically, suppose we want to compute the Lie algebra element, $\boldsymbol{\xi}_{ac} \in \mathfrak{g}$, that is generated by taking the logarithm of the product of the exponential of two Lie algebra elements $\boldsymbol{\xi}_{ab}, \boldsymbol{\xi}_{bc} \in \mathfrak{g}$. The *Baker-Campbell-Hausdorff (BCH)* formula describes this relationship purely in the Lie algebra space without requiring the application of the exponential or the logarithm:

$$\begin{aligned} \boldsymbol{\xi}_{ac} &= \log(\exp(\boldsymbol{\xi}_{ab})\exp(\boldsymbol{\xi}_{bc})) \\ &= \boldsymbol{\xi}_{ab} + \boldsymbol{\xi}_{bc} + \frac{1}{2}[\boldsymbol{\xi}_{ab}, \boldsymbol{\xi}_{bc}] + \\ &\quad + \frac{1}{12}([\boldsymbol{\xi}_{ab}, [\boldsymbol{\xi}_{ab}, \boldsymbol{\xi}_{bc}]] + [\boldsymbol{\xi}_{bc}, [\boldsymbol{\xi}_{bc}, \boldsymbol{\xi}_{ab}]]) + \cdots, \end{aligned} \tag{11}$$

where $[\cdot, \cdot]$ is the Lie bracket of $\mathfrak{g}$ [17, (7.18)]. For brevity, in the special case where we consider elements in $\mathrm{SE}(3)$, we adopt the notation of [1]:

$$\boldsymbol{\xi}_i^\curlywedge := \begin{bmatrix} \boldsymbol{\rho}_i \\ \boldsymbol{\phi}_i \end{bmatrix}^\curlywedge = \begin{bmatrix} \boldsymbol{\phi}_i^\wedge & \boldsymbol{\rho}_i^\wedge \\ \mathbf{0} & \boldsymbol{\phi}_i^\wedge \end{bmatrix}, \tag{12}$$

which allows us to write the BCH formula as:

$$\begin{aligned} \boldsymbol{\xi}_{ac} = \boldsymbol{\xi}_{ab} + \boldsymbol{\xi}_{bc} + \frac{1}{2}\boldsymbol{\xi}_{ab}^\curlywedge\boldsymbol{\xi}_{bc} + \frac{1}{12}\boldsymbol{\xi}_{ab}^\curlywedge\boldsymbol{\xi}_{ab}^\curlywedge\boldsymbol{\xi}_{bc} + \frac{1}{12}\boldsymbol{\xi}_{bc}^\curlywedge\boldsymbol{\xi}_{bc}^\curlywedge\boldsymbol{\xi}_{ab} \\ - \frac{1}{24}\boldsymbol{\xi}_{bc}^\curlywedge\boldsymbol{\xi}_{ab}^\curlywedge\boldsymbol{\xi}_{ab}^\curlywedge\boldsymbol{\xi}_{bc} + \cdots. \end{aligned} \tag{13}$$

E. Defining Random Variables over Poses

Finally, as discussed in [1], one can define random variables for $\mathrm{SE}(3)$ according to:[1]

$$\mathbf{T}_\ell := \exp(\boldsymbol{\xi}_\ell^\wedge)\bar{\mathbf{T}}_\ell \tag{14}$$

where $\bar{\mathbf{T}}_\ell \in \mathrm{SE}(3)$ is a 'large' noise free value and $\boldsymbol{\xi}_\ell \in \mathbb{R}^6$ is a 'small' noisy perturbation (using the nomenclature of [1]). Two examples of this noisy perturbation are depicted in the middle row of Fig. 1. By defining $\boldsymbol{\xi}_\ell$ to be a zero-mean Gaussian random variable $\boldsymbol{\xi}_\ell \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\ell)$ in the Lie algebra, we induce a probability distribution function over $\mathrm{SE}(3)$ that

[1]In this paper we follow the convention of using the exponential map to perturb a mean group element on the left side. A potential alternative would be to multiply on the right. The primary difference between these two forms is the frame in which the perturbation is applied [29]. All of the results in the paper have equivalent derivations for right handed perturbation as well, however, to save space, in this article we perform multiplication on the left to follow the convention used in [1].

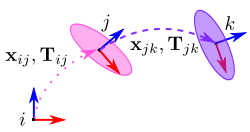
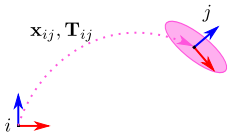
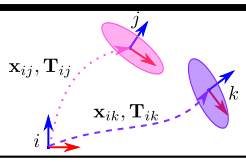
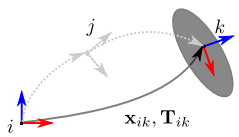
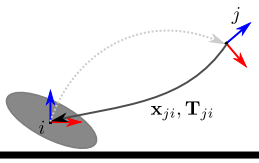
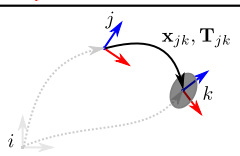
	Pose Composition	Pose Inverse	Relative Pose
Input			
Output			
SSC [5]	$\mathbf{x}_{ab} = [x_{ab}, y_{ab}, z_{ab}, \phi_{ab}, \theta_{ab}, \psi_{ab}]^\top$		
	$\mathbf{x} = [\mathbf{x}_{ij}^\top, \mathbf{x}_{jk}^\top]^\top \sim \mathcal{N}(\hat{\mathbf{x}}, \hat{\mathbf{\Sigma}})$	$\mathbf{x}_{ij} \sim \mathcal{N}(\hat{\mathbf{x}}_{ij}, \mathbf{\Sigma}_{\mathbf{x}_{ij}})$	$\mathbf{x} = [\mathbf{x}_{ij}^\top, \mathbf{x}_{ik}^\top]^\top \sim \mathcal{N}(\hat{\mathbf{x}}, \hat{\mathbf{\Sigma}})$
	$\hat{\mathbf{x}}_{ik} = \hat{\mathbf{x}}_{ij} \oplus \hat{\mathbf{x}}_{jk}$ $\mathbf{\Sigma}_{\mathbf{x}_{ik}} \approx J_{\oplus}(\hat{\mathbf{x}}) \mathbf{\Sigma} J_{\oplus}(\hat{\mathbf{x}})^\top$	$\hat{\mathbf{x}}_{ji} = \ominus \hat{\mathbf{x}}_{ij}$ $\mathbf{\Sigma}_{\mathbf{x}_{ji}} \approx J_{\ominus}(\hat{\mathbf{x}}_{ij}) \mathbf{\Sigma}_{\mathbf{x}_{ij}} J_{\ominus}(\hat{\mathbf{x}}_{ij})^\top$	$\hat{\mathbf{x}}_{jk} = (\ominus \hat{\mathbf{x}}_{ij}) \oplus \hat{\mathbf{x}}_{ik}$ $\mathbf{\Sigma}_{\mathbf{x}_{jk}} \approx J_{\ominus\oplus}(\hat{\mathbf{x}}) \mathbf{\Sigma} J_{\ominus\oplus}(\hat{\mathbf{x}})^\top$
[1]	$\mathbf{T}_{ab} = \exp(\boldsymbol{\xi}_{ab}^\wedge) \bar{\mathbf{T}}_{ab} \qquad \bar{\mathbf{T}}_{ab} \in \mathrm{SE}(3), \quad \boldsymbol{\xi}_{ab}^\wedge \in \mathfrak{se}(3)$		
	$\boldsymbol{\xi}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{ij})$ $\boldsymbol{\xi}_{jk} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{jk})$	–	–
	$\bar{\mathbf{T}}_{ik} \triangleq \bar{\mathbf{T}}_{ij} \bar{\mathbf{T}}_{jk}$ $\mathbf{\Sigma}_{ik} \approx \mathbf{\Sigma}_{ij} + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}} \mathbf{\Sigma}_{jk} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}$		
Proposed	$\mathbf{T}_{ab} = \exp(\boldsymbol{\xi}_{ab}^\wedge) \bar{\mathbf{T}}_{ab} \qquad \bar{\mathbf{T}}_{ab} \in \mathrm{SE}(3), \quad \boldsymbol{\xi}_{ab}^\wedge \in \mathfrak{se}(3)$		
	$\boldsymbol{\xi} = [\boldsymbol{\xi}_{ij}^\top, \boldsymbol{\xi}_{jk}^\top]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ $\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{ij} & \mathbf{\Sigma}_{ij,jk} \\ \mathbf{\Sigma}_{ij,jk}^\top & \mathbf{\Sigma}_{jk} \end{bmatrix}$	$\boldsymbol{\xi}_{ij} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{ij})$	$\boldsymbol{\xi} = [\boldsymbol{\xi}_{ij}^\top, \boldsymbol{\xi}_{ik}^\top]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ $\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Sigma}_{ij} & \mathbf{\Sigma}_{ij,ik} \\ \mathbf{\Sigma}_{ij,ik}^\top & \mathbf{\Sigma}_{ik} \end{bmatrix}$
	$\bar{\mathbf{T}}_{ik} \triangleq \bar{\mathbf{T}}_{ij} \bar{\mathbf{T}}_{jk}$ $\mathbf{\Sigma}_{ik} \approx \mathbf{\Sigma}_{ij} + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}} \mathbf{\Sigma}_{jk} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}} +$ $+ \mathbf{\Sigma}_{ij,jk} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}^\top + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}} \mathbf{\Sigma}_{ij,jk}^\top$	$\bar{\mathbf{T}}_{ji} \triangleq \bar{\mathbf{T}}_{ij}^{-1}$ $\mathbf{\Sigma}_{ji} \approx \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \mathbf{\Sigma}_{ij} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^\top$	$\bar{\mathbf{T}}_{jk} \triangleq \bar{\mathbf{T}}_{ij}^{-1} \bar{\mathbf{T}}_{ik}$ $\mathbf{\Sigma}_{jk} \approx$ $\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \mathbf{\Sigma}_{ij} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^\top + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \mathbf{\Sigma}_{ik} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^\top +$ $- \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \mathbf{\Sigma}_{ij,ik} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^\top - \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \mathbf{\Sigma}_{ij,ik}^\top \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^\top$

Fig. 3: Summary of the pose composition, pose inverse, and relative pose operations and their corresponding uncertainty propagation methods as proposed by Smith et al. [5], Barfoot and Furgale [1], and in this paper. The indices i, j, and k correspond to specific coordinate frames of the robotic vehicle at different time steps or locations. SSC [5] parameterizes the pose of frame b with respect to frame a using a vector of Euler angle and translation parameters $\mathbf{x}_{ab}$. They then model this vector of parameters as being drawn from a multivariate Gaussian distribution with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{\Sigma}$. Under this model, they derive first order uncertainty propagation formulas for the pose composition, pose inverse, and relative pose operations. However, this coordinate based parameterization is unable to accurately model pose uncertainty because the parameter vector $\mathbf{x}_{ab}$ is not truly Gaussian. Barfoot and Furgale [1] instead parameterize the pose of frame b with respect to frame a, using a mean element of the Special Euclidean group, $\bar{\mathbf{T}}_{ab}$, and an uncertain perturbation or noise parameter $\boldsymbol{\xi}_{ab}^\wedge$ defined in the Lie algebra $\mathfrak{se}(3)$. This enables them to model $\boldsymbol{\xi}_{ab}$ using a Gaussian distribution and accurately take into account the non-linear structure of the group. However, [1] assumes the poses are independent and focuses primarily on the pose composition operation. The **primary contribution** of this paper is the extension of the method presented by Barfoot and Furgale [1] to jointly correlated poses as well as the derivation of the pose inverse and relative pose operations while taking advantage of the Lie algebra to characterize uncertainty.

is parameterized with a mean $\bar{\mathbf{T}}_\ell \in \mathrm{SE}(3)$ and a covariance $\mathbf{\Sigma}_\ell$ defined in the Lie algebra [1].

We use these properties in Sections V, VI, VII, and VIII to propose a framework for modeling jointly correlated poses and to derive uncertainty propagation formulas for the operations in Fig. 3. However, we first review the SSC coordinate based method.

IV. Review of SSC

The stochastic map, proposed by Smith et al. [5], consists of multiple uncertain spatial relationships that are treated as jointly Gaussian multivariate random variables and parameterized using a mean vector of positions and Euler angles and an associated covariance matrix. Smith et al. [5] also proposes three operations that can extract information from this map that may not have been directly estimated. These three operations are pose composition (head-to-tail in [5]), pose inverse, and relative pose estimation (tail-to-tail in [5]), as shown in Fig. 3. This section reviews the pose representation and SSC operations formulated in [5].

A. Pose Representation and the Stochastic Map

Under SSC notation, the relative transformation between the coordinate frame i and j, or the pose of coordinate frame j with respect to frame i, is denoted by

$$\mathbf{x}_{ij} = [x_{ij}, y_{ij}, z_{ij}, \phi_{ij}, \theta_{ij}, \psi_{ij}]^\top, \tag{15}$$

where x_{ij}, y_{ij}, and z_{ij} are position coordinates and ϕ_{ij}, θ_{ij}, and ψ_{ij} are Euler angles that encode the position and orientation of the coordinate frame j with respect to the frame i. However, because each relative transformation is derived from noisy measurements, our estimate is uncertain and we do not know the true value of $\mathbf{x}_{ij}$. Instead, we track a nominal mean value of its parameters and an associated 6×6 covariance matrix, $\hat{\mathbf{x}}_{ij}$ and $\mathbf{\Sigma}_{\mathbf{x}_{ij}}$, respectively.

The *stochastic map*, proposed in [5], treats all of the uncertain transformations as jointly Gaussian multivariate random variables by stacking them into a single state vector $\mathbf{x}$ and tracking the mean and covariance of that state. Assuming we have n relative transformations we want to track, if we index them from $a = 1, \ldots, n$, then $\mathbf{x}$ and its associated mean and covariance are:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}, \hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \vdots \\ \hat{\mathbf{x}}_n \end{bmatrix}, \hat{\mathbf{\Sigma}} = \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{x}_1} & \cdots & \mathbf{\Sigma}_{\mathbf{x}_1\mathbf{x}_n} \\ \vdots & \ddots & \vdots \\ \mathbf{\Sigma}_{\mathbf{x}_n\mathbf{x}_1} & \cdots & \mathbf{\Sigma}_{\mathbf{x}_n} \end{bmatrix} \tag{16}$$

where $\mathbf{x}$ is a vector of length $6n$, $\mathbf{\Sigma}_{\mathbf{x}_a}$ is the 6×6 covariance matrix of the relative transformation $\mathbf{x}_a$ and the off diagonal blocks of the form $\mathbf{\Sigma}_{\mathbf{x}_a\mathbf{x}_b}$ are the respective 6×6 cross covariance matrices.

B. Pose Composition (Head-to-Tail)

Suppose we are given a noisy observation of a robot's relative pose between time steps i and j ($\mathbf{x}_{ij}$) and another observation of its relative pose between time steps j and k ($\mathbf{x}_{jk}$), we may want to calculate the relative pose between time steps i and k ($\mathbf{x}_{ik}$) by composing the observations $\mathbf{x}_{ij}$ and $\mathbf{x}_{jk}$ (see Fig. 3). The SSC *head-to-tail* operation is a nonlinear function $f_\oplus : \mathbb{R}^6 \times \mathbb{R}^6 \to \mathbb{R}^6$ that takes the parameter vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{jk}$ as input and outputs the parameter vector $\mathbf{x}_{ik}$ that results from composing the respective homogeneous transformation matrices. The $\oplus$ operator denotes this operation:

$$\mathbf{x}_{ik} \triangleq \mathbf{x}_{ij} \oplus \mathbf{x}_{jk} = f_\oplus(\mathbf{x}_{ij}, \mathbf{x}_{jk}). \tag{17}$$

The mean and covariance of the resulting pose are estimated up-to first order as

$$\hat{\mathbf{x}}_{ik} = \hat{\mathbf{x}}_{ij} \oplus \hat{\mathbf{x}}_{jk} \tag{18}$$

and

$$\mathbf{\Sigma}_{\mathbf{x}_{ik}} \approx J_\oplus(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{jk})\ \hat{\mathbf{\Sigma}}\ J_\oplus(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{jk})^\top \tag{19}$$

where $J_\oplus(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{jk})$ is the Jacobian of $f_\oplus$ at $\hat{\mathbf{x}}_{ij}$ and $\hat{\mathbf{x}}_{jk}$, and

$$\hat{\mathbf{\Sigma}} = \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{x}_{ij}} & \mathbf{\Sigma}_{\mathbf{x}_{ij}\mathbf{x}_{jk}} \\ \mathbf{\Sigma}_{\mathbf{x}_{ij}\mathbf{x}_{jk}}^\top & \mathbf{\Sigma}_{\mathbf{x}_{jk}} \end{bmatrix}. \tag{20}$$

C. Pose Inverse

Suppose we are given a robotic vehicle that is tasked with navigating through an *a priori* unknown environment after which it must return to the origin location. While it is common to represent the robotic vehicle's current location with respect to the origin using estimation theory, it may be useful to instead characterize the pose of the origin with respect to the local robot coordinate frame. Formally, given an uncertain estimate of the pose of coordinate frame j with respect to frame i ($\mathbf{x}_{ij}$), we want to determine the pose of frame i with respect to frame j ($\mathbf{x}_{ji}$). This amounts to finding the vector $\hat{\mathbf{x}}_{ji}$ that corresponds to the inverse of $\hat{\mathbf{x}}_{ij}$ (in terms of homogeneous transformation matrices) and representing the uncertainty with respect to this new frame of reference.

The SSC *pose inverse* operation is a nonlinear function $f_\ominus : \mathbb{R}^6 \to \mathbb{R}^6$ takes a pose and computes the inverse of the pose as a homogeneous transformation matrix:

$$\mathbf{x}_{ji} \triangleq \ominus\mathbf{x}_{ij} = f_\ominus(\mathbf{x}_{ij}). \tag{21}$$

The mean and covariance of the resulting pose are estimated up-to first order as

$$\hat{\mathbf{x}}_{ji} = \ominus\hat{\mathbf{x}}_{ij} \tag{22}$$

and

$$\mathbf{\Sigma}_{\mathbf{x}_{ji}} \approx J_\ominus(\hat{\mathbf{x}}_{ij})\ \mathbf{\Sigma}_{\mathbf{x}_{ij}}\ J_\ominus(\hat{\mathbf{x}}_{ij})^\top \tag{23}$$

where $J_\ominus(\hat{\mathbf{x}}_{ij})$ is the Jacobian of $f_\ominus$ at $\hat{\mathbf{x}}_{ij}$.

D. Relative Pose (Tail-to-Tail)

Finally, the SSC *tail-to-tail* operation takes uncertain estimates of two coordinate frames with respect to a single origin frame and evaluates the relative pose between them. More succinctly, given $\mathbf{x}_{ij}$ and $\mathbf{x}_{ik}$, find $\mathbf{x}_{jk}$. This operation is useful when using a method such as pose graph SLAM for navigation since the robot pose at multiple time steps is often estimated with respect to a single fixed coordinate frame.

The SSC *relative pose* operation is a nonlinear function $f_{\ominus\oplus} : \mathbb{R}^6 \times \mathbb{R}^6 \to \mathbb{R}^6$ that is defined by first applying the inverse and then the head-to-tail operation:

$$\mathbf{x}_{jk} \triangleq (\ominus\mathbf{x}_{ij}) \oplus \mathbf{x}_{ik} = f_{\ominus\oplus}(\mathbf{x}_{ij}, \mathbf{x}_{ik}). \tag{24}$$

The mean and covariance of the resulting pose up-to first order are estimated as

$$\hat{\mathbf{x}}_{jk} = (\ominus \hat{\mathbf{x}}_{ij}) \oplus \hat{\mathbf{x}}_{ik} \tag{25}$$

and

$$\boldsymbol{\Sigma}_{\mathbf{x}_{jk}} \approx J_{\ominus\oplus}(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{ik})\ \hat{\boldsymbol{\Sigma}}\ J_{\ominus\oplus}(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{ik})^\top \tag{26}$$

where $J_{\ominus\oplus}(\hat{\mathbf{x}}_{ij}, \hat{\mathbf{x}}_{ik})$ is the Jacobian of $f_{\ominus\oplus}$ at $\hat{\mathbf{x}}_{ij}$ and $\hat{\mathbf{x}}_{ik}$, and

$$\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{x}_{ij}} & \boldsymbol{\Sigma}_{\mathbf{x}_{ij}\mathbf{x}_{ik}} \\ \boldsymbol{\Sigma}^\top_{\mathbf{x}_{ij}\mathbf{x}_{ik}} & \boldsymbol{\Sigma}_{\mathbf{x}_{ik}} \end{bmatrix}. \tag{27}$$

V. Jointly Characterizing Uncertainty in the Lie algebra

While the SSC operations proposed in [5] have been used widely since they were introduced, over the past decade Lie algebra based methods have been shown to provide a more accurate characterization of uncertainty [1, 8]. However, while recent years have seen an increase in use of Lie algebra based methods for uncertainty propagation [30–32], existing methods assume that individual measurements are independent [1], which may not be the case when the underlying pose estimates are derived from a SLAM solution. We now describe how this assumption can be dropped and poses can be modeled as jointly correlated within the Lie algebra space.

Assume that we have n uncertain poses $\{\mathbf{T}_1, \ldots, \mathbf{T}_n\}$, each of which is defined according to (14). If we assume that the poses are statistically independent, then we can parameterize this distribution of poses with the set of associated mean and covariance matrices $\{\bar{\mathbf{T}}_1, \boldsymbol{\Sigma}_1, \ldots, \bar{\mathbf{T}}_n, \boldsymbol{\Sigma}_n\}$. However, if the uncertainty associated with the set of poses is correlated, then this can be modeled by concatenating the set of perturbation vectors $\{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n\}$ into a single vector $\boldsymbol{\xi}_{1:n} \in \mathbb{R}^{6n}$ and represent the uncertainty of the distribution using a single covariance matrix:

$$\boldsymbol{\xi}_{1:n} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \vdots \\ \boldsymbol{\xi}_n \end{bmatrix}, \quad \boldsymbol{\Sigma}_{1:n} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \ldots, & \boldsymbol{\Sigma}_{1,n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}^\top_{1,n} & \ldots & \boldsymbol{\Sigma}_n \end{bmatrix}, \tag{28}$$

where $\boldsymbol{\xi}_{1:n} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{1:n})$. Thus, we can parameterize the distribution of n poses using the set of each of their means

$$\bar{\mathbf{T}}_{1:n} = \{\bar{\mathbf{T}}_1, \ldots, \bar{\mathbf{T}}_n\} \tag{29}$$

and the covariance matrix $\boldsymbol{\Sigma}_{1:n}$. Note that this only extends since we define the uncertainty in the Lie algebra. The next three sections describe how to apply the composition, inverse, and relative pose operations on uncertain poses represented in this manner.

VI. Jointly Distributed Pose Composition

This section describes how to compose two uncertain poses who's associated perturbation vectors are jointly Gaussian in the Lie algebra as described in the last section. This is the Lie group-based or coordinate free equivalent to the SSC head-to-tail operation described in Section IV-B.

A. Pose Composition Operation Derivation

Suppose we have two uncertain poses $\mathbf{T}_{ij}$ and $\mathbf{T}_{jk}$ with perturbations $\boldsymbol{\xi}_{ij}$ and $\boldsymbol{\xi}_{jk}$ that are jointly Gaussian in the Lie algebra with covariance matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{ij} & \boldsymbol{\Sigma}_{ij,jk} \\ \boldsymbol{\Sigma}^\top_{ij,jk} & \boldsymbol{\Sigma}_{jk} \end{bmatrix}.$$

Our goal is to find the mean and covariance of $\mathbf{T}_{ik}$, $\{\bar{\mathbf{T}}_{ik}, \boldsymbol{\Sigma}_{ik}\}$. Under the standard group multiplication operation:

$$\mathbf{T}_{ik} = \mathbf{T}_{ij}\mathbf{T}_{jk}. \tag{30}$$

Following the random variable definition in (14) and using the property described in (10),

$$\begin{aligned} \exp(\boldsymbol{\xi}^\wedge_{ik})\bar{\mathbf{T}}_{ik} &= \exp(\boldsymbol{\xi}^\wedge_{ij})\bar{\mathbf{T}}_{ij}\exp(\boldsymbol{\xi}^\wedge_{jk})\bar{\mathbf{T}}_{jk} \\ &= \exp(\boldsymbol{\xi}^\wedge_{ij})\exp((\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}\boldsymbol{\xi}_{jk})^\wedge)\bar{\mathbf{T}}_{ij}\bar{\mathbf{T}}_{jk} \end{aligned} \tag{31}$$

where $\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}$ is the matrix form of the adjoint action of $\bar{\mathbf{T}}_{ij}$ on $\mathfrak{se}(3)$. Letting

$$\bar{\mathbf{T}}_{ik} \triangleq \bar{\mathbf{T}}_{ij}\bar{\mathbf{T}}_{jk}, \tag{32}$$

gives us

$$\exp(\boldsymbol{\xi}^\wedge_{ik}) = \exp(\boldsymbol{\xi}^\wedge_{ij})\exp((\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}\boldsymbol{\xi}_{jk})^\wedge). \tag{33}$$

We can now use the Baker-Campbell-Housdorff (BCH) formula (13) to show that

$$\begin{aligned} \boldsymbol{\xi}_{ik} = \boldsymbol{\xi}_{ij} + \boldsymbol{\xi}'_{jk} + \frac{1}{2}\boldsymbol{\xi}^\curlywedge_{ij}\boldsymbol{\xi}'_{jk} + \frac{1}{12}\boldsymbol{\xi}^\curlywedge_{ij}\boldsymbol{\xi}^\curlywedge_{ij}\boldsymbol{\xi}'_{jk} + \frac{1}{12}\boldsymbol{\xi}'^\curlywedge_{jk}\boldsymbol{\xi}'^\curlywedge_{jk}\boldsymbol{\xi}_{ij} + \\ - \frac{1}{24}\boldsymbol{\xi}'^\curlywedge_{jk}\boldsymbol{\xi}^\curlywedge_{ij}\boldsymbol{\xi}^\curlywedge_{ij}\boldsymbol{\xi}'_{jk} + \ldots \end{aligned} \tag{34}$$

where $\boldsymbol{\xi}'_{jk} = \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}\boldsymbol{\xi}_{jk}$. Computing the covariance $\boldsymbol{\Sigma}$ amounts to evaluating $E[\boldsymbol{\xi}_{ik}\boldsymbol{\xi}^\top_{ik}]$, where $E[\cdot]$ signifies the expected value of a given random variable. Multiplying out up-to fourth order, we get:

$$E[\boldsymbol{\xi}_{ik}\boldsymbol{\xi}^\top_{ik}] \approx E\Bigg[\underbrace{\boldsymbol{\xi}_{ij}\boldsymbol{\xi}^\top_{ij} + \boldsymbol{\xi}'_{jk}\boldsymbol{\xi}'^\top_{jk}}_{\text{2nd Order Diag. Terms}} + \underbrace{\boldsymbol{\xi}_{ij}\boldsymbol{\xi}'^\top_{jk} + \boldsymbol{\xi}'_{jk}\boldsymbol{\xi}^\top_{ij}}_{\text{2nd Order Cross Terms}} + \tag{35}$$

$$+ \frac{1}{12}((\boldsymbol{\xi}^\curlywedge_{ij}\boldsymbol{\xi}^\curlywedge_{ij})(\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}'^\top_{jk}) + (\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}'^\top_{jk})(\boldsymbol{\xi}^\curlywedge_{ij}\boldsymbol{\xi}^\curlywedge_{ij})^\top + \\ + \quad (\boldsymbol{\xi}'^\curlywedge_{jk}\boldsymbol{\xi}'^\curlywedge_{jk})(\boldsymbol{\xi}_{ij}\boldsymbol{\xi}^\top_{ij}) + (\boldsymbol{\xi}_{ij}\boldsymbol{\xi}^\top_{ij})(\boldsymbol{\xi}'^\curlywedge_{jk}\boldsymbol{\xi}'^\curlywedge_{jk})^\top) + \tag{36}$$

$$+ \underbrace{\frac{1}{4}\boldsymbol{\xi}^\curlywedge_{ij}(\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}'^\top_{jk})\boldsymbol{\xi}^{\curlywedge\top}_{ij} +}_{\text{4th Order Diagonal Terms}}$$

$$+ \frac{1}{12}((\boldsymbol{\xi}_{ij}\boldsymbol{\xi}'^\top_{jk})(\boldsymbol{\xi}^{\curlywedge\top}_{ij}\boldsymbol{\xi}^{\curlywedge\top}_{ij}) + (\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}^\top_{ij})(\boldsymbol{\xi}'^{\curlywedge\top}_{jk}\boldsymbol{\xi}'^{\curlywedge\top}_{jk}) + \tag{37}$$

$$+ \underbrace{(\boldsymbol{\xi}^\curlywedge_{ij}\boldsymbol{\xi}^\curlywedge_{ij})(\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}^\top_{ij}) + (\boldsymbol{\xi}'^\curlywedge_{jk}\boldsymbol{\xi}'^\curlywedge_{jk})(\boldsymbol{\xi}_{ij}\boldsymbol{\xi}'^\top_{jk}))}_{\text{4th Order Cross Terms}} \Bigg].$$

This derivation is identical to the one proposed in [1] until the last step. Since [1] assumes the individual poses are independent, the cross terms in (35) and (37) are zero and the evaluation of the 4th order terms in (36) is simplified. If this

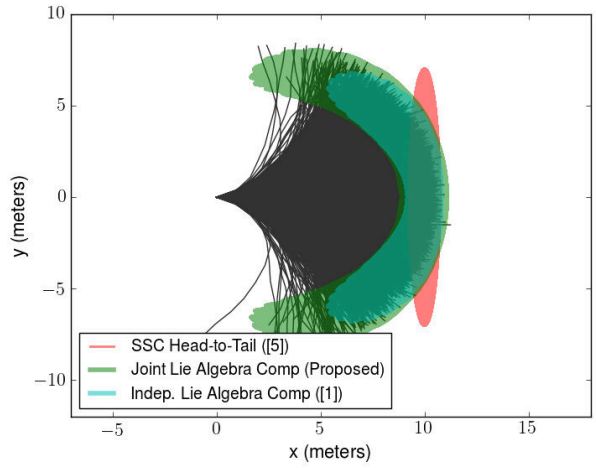


Fig. 4: Plots of 10000 sample trajectories (shown in black), each made up of a sequence of 10 noisy pose transformations starting from the same point. The 95% likely uncertainty ellipse predicted by first order uncertainty propagation through the SSC head-to-tail operation is shown in red overlayed on top of the sample trajectories, while a representation of the flattened 95% likely uncertainty position ellipsoids predicted by the Lie algebra pose composition methods when correlation is and is not taken into account are overlayed in transparent green and cyan, respectively.

assumption is true and $\boldsymbol{\xi}_{ij}$ and $\boldsymbol{\xi}'_{jk}$ are independent of one another, as is the case when $\mathbf{T}_{ij}$ and $\mathbf{T}_{jk}$ are independent measurements of consecutive robot motion, then the cross terms in (36) and the covariance $\boldsymbol{\Sigma}$ can be calculated up to 4th order as described in [1]. However, if the poses $\mathbf{T}_{ij}$ and $\mathbf{T}_{jk}$ are correlated, as is often the case when they are derived from the solution of a MLE problem such as Pose Graph SLAM or in the case of wheel slip, the cross terms must be included and evaluation of the fourth order terms in (36) and (37) becomes more difficult[2]. If the correlation is not taken into account, then the result of the pose composition operation under-approximates the true distribution and consistency is lost as shown in Fig. 4.

A first order estimate of covariance (second order in the perturbation variables) can be obtained be evaluating

$$\begin{aligned} E[\boldsymbol{\xi}_{ik}\boldsymbol{\xi}_{ik}^\top] \approx & E[\boldsymbol{\xi}_{ij}\boldsymbol{\xi}_{ij}^\top] + E[\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}'^\top_{jk}] + \\ & + E[\boldsymbol{\xi}_{ij}\boldsymbol{\xi}'^\top_{jk}] + E[\boldsymbol{\xi}'_{jk}\boldsymbol{\xi}_{ij}^\top], \end{aligned} \tag{38}$$

resulting in

$$\begin{aligned} \boldsymbol{\Sigma}_{ik} \approx & \boldsymbol{\Sigma}_{ij} + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}} \boldsymbol{\Sigma}_{jk} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}^\top + \\ & + \boldsymbol{\Sigma}_{ij,jk} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}}^\top + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}} \boldsymbol{\Sigma}_{ij,jk}^\top. \end{aligned} \tag{39}$$

Thus, first order pose composition on the Lie algebra can be performed for correlated poses using (32) for mean propagation and (39) for covariance propagation. As noted previously, this is equivalent to the first order method presented in [1] with the exception of the two additional cross terms in (39) needed to model cross correlation.

B. Accurately Modeling Group Structure

While both the SSC operations presented in [5] and the Lie algebra based methods presented in this paper (and Σ_{2nd} in [1, (55)]) are first order approximation methods (in terms of covariance), characterizing rotation uncertainty in the Lie algebra space results in a more accurate characterization of pose uncertainty. This is because Euler angle parameterization is a chart and does not cover the entire manifold whereas the Lie algebra inherently captures the group structure and can model any arbitrary group element. In addition, since the Lie algebra is a vector space [27], modeling the uncertainty as a Gaussian distribution is convenient and well-defined.

To demonstrate this, we performed an experiment similar to the one proposed in [1]. We generated a sequence of N noisy pose transformations of the form $\mathbf{T}_{ab} = \exp(\boldsymbol{\xi}_{ab}^\wedge)\bar{\mathbf{T}}_{ab}$, with

$$\boldsymbol{\xi}_{ab} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{ab}), \tag{40}$$

$$\boldsymbol{\Sigma}_{ab} = \mathrm{diag}([0.001\sigma_t, 1e^{-5}\sigma_t, 1e^{-5}, 1e^{-5}, 1e^{-5}, 0.003\sigma_r]), \tag{41}$$

and

$$\bar{\mathbf{T}}_{ab} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{42}$$

where σ_t and σ_r are scaling parameters for the translation and rotation noise respectively. In addition, each consecutive perturbation variable $\boldsymbol{\xi}_{ab}$ was drawn such that it was correlated with the perturbation variable before it with correlation coefficient ρ.

We then composed these transformations end-to-end and repeated the process to generate 10000 sample trajectories. These Monte-Carlo simulation results were then used to evaluate the uncertainty predicted by the SSC head-to-tail operation as well as the Lie-algebra based pose composition method derived in this section, both with and without correlation being taken into account. A top down view of one such experiment with $N = 10$, $\sigma_t = 5$, $\sigma_r = 5$, and $\rho = 0.4$ is shown in Fig. 4.

As expected, the Lie algebra based method significantly outperforms SSC head-to-tail because it takes into account the structure of the $\mathrm{SE}(3)$ group. In addition, it is easily seen that dropping the cross covariance terms results in an under-approximation of the true covariance, if positive correlation is indeed present. A more thorough investigation of this experiment is described in Section X-A.

In the next two sections, we provide a derivation of the inverse and relative pose operations on the Lie algebra, which as far as we know have not been previously published.

VII. The Pose Inverse Operation

The pose inverse operation corresponds to a change in reference frame. Given an uncertain pose distribution $\mathbf{T}_{ij}$ that represents the pose of coordinate frame j with respect to frame i, we want to find the inverse $\mathbf{T}_{ji}$ that represents the pose of coordinate frame i with respect to frame j. The distribution of an inverse pose can be derived as follows [33]:

$$\begin{aligned} \mathbf{T}_{ji} &= \mathbf{T}_{ij}^{-1} \\ &= (\exp(\boldsymbol{\xi}_{ij}^\wedge)\bar{\mathbf{T}}_{ij})^{-1} \\ &= \bar{\mathbf{T}}_{ij}^{-1} \exp(-\boldsymbol{\xi}_{ij}^\wedge) \\ &= \exp((-\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}\boldsymbol{\xi}_{ij})^\wedge)\bar{\mathbf{T}}_{ij}^{-1} \\ &= \exp(\boldsymbol{\xi}'_{ij})\bar{\mathbf{T}}_{ij}^{-1} \end{aligned} \tag{43}$$

[2]If increased accuracy is needed then Isserlis Theorem can be applied to evaluate the fourth order terms in (36) and (37).

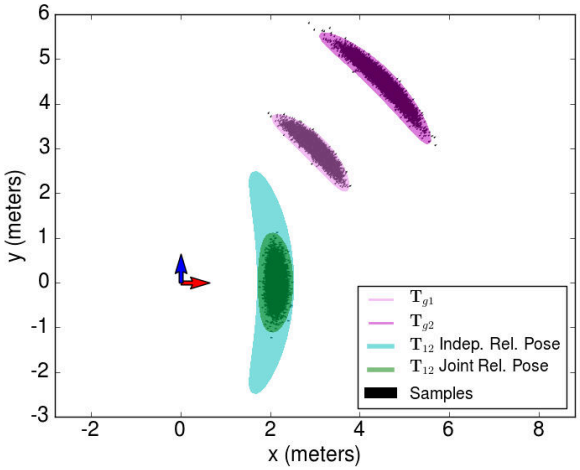


Fig. 5: Dropping the correlation terms in (39) and (51) can lead to under/over-estimation of uncertainty. In this example, we used our proposed method to estimate the relative pose $\mathbf{T}_{12}$ between two correlated poses $\mathbf{T}_{g1}$ and $\mathbf{T}_{g2}$. A flattened view of the predicted 95% likely uncertainty ellipsoids that result when correlation is ignored ((51) without the cross terms) and taken into account ((51) with all terms) are shown in cyan and green respectively. This plot was with $\alpha = 1$.

where $\boldsymbol{\xi}'_{ij}$ is the original perturbation $\boldsymbol{\xi}_{ij}$ transformed by the negative adjoint of $\bar{\mathbf{T}}_{ij}^{-1}$. Due to the linearity of the adjoint operation, we can represent the inverse distribution $\mathbf{T}_{ji}$ with mean and covariance

$$\bar{\mathbf{T}}_{ji} = \bar{\mathbf{T}}_{ij}^{-1} \tag{44}$$

and

$$\boldsymbol{\Sigma}_{ji} = \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \boldsymbol{\Sigma}_{ij} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^{\top}. \tag{45}$$

VIII. Finding the Relative Transformation Between Jointly Distributed Poses

This section combines the derivations from the previous two sections to formulate a first order method for characterizing the uncertainty of the relative pose operation or the coordinate-free equivalent to the SSC tail-to-tail operation described in Section IV-D. As far as we can tell, this is the first time this operation has been published while using the Lie algebra to characterize uncertainty as defined in [1].

A. Relative Pose Operation Derivation

Given uncertain estimates of two coordinate frames with respect to a common base frame, our aim is estimate the relative transformation between the two poses. For example, given possibly correlated uncertain transformations $\mathbf{T}_{ij}$ and $\mathbf{T}_{ik}$ representing the poses of coordinate frames j and k with respect to frame i, we want to find the mean $\bar{\mathbf{T}}_{jk}$ and covariance $\boldsymbol{\Sigma}_{jk}$ that parameterize the pose uncertainty of frame k with respect to frame j.

We start by assuming that the $\bar{\mathbf{T}}_{ij}$ and $\bar{\mathbf{T}}_{ik}$ are known and that the associated perturbations $\boldsymbol{\xi}_{ij}$ and $\boldsymbol{\xi}_{ik}$ are jointly correlated in the Lie algebra with known covariance

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{ij} & \boldsymbol{\Sigma}_{ij,ik} \\ \boldsymbol{\Sigma}_{ij,ik}^{\top} & \boldsymbol{\Sigma}_{ik} \end{bmatrix}.$$

Under the standard group inverse and multiplication actions, the following must hold:

$$\mathbf{T}_{jk} = \mathbf{T}_{ij}^{-1} \mathbf{T}_{ik}. \tag{46}$$

Expanding (46) using the random variable definition in (14) and the derivation in (43) results in

$$\begin{aligned} \exp(\boldsymbol{\xi}_{jk}^{\wedge}) \bar{\mathbf{T}}_{jk} &= \bar{\mathbf{T}}_{ij}^{-1} \exp(-\boldsymbol{\xi}_{ij}^{\wedge}) \exp(\boldsymbol{\xi}_{ik}^{\wedge}) \bar{\mathbf{T}}_{ik} \\ &= \exp((-\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \boldsymbol{\xi}_{ij})^{\wedge}) \bar{\mathbf{T}}_{ij}^{-1} \exp(\boldsymbol{\xi}_{ik}^{\wedge}) \bar{\mathbf{T}}_{ik} \\ &= \exp(\boldsymbol{\xi}_{ij}'^{\wedge}) \exp(\boldsymbol{\xi}_{ik}'^{\wedge}) \bar{\mathbf{T}}_{ij}^{-1} \bar{\mathbf{T}}_{ik}, \end{aligned} \tag{47}$$

where $\boldsymbol{\xi}'_{ij} = -\mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \boldsymbol{\xi}_{ij}$ and $\boldsymbol{\xi}'_{ik} = \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \boldsymbol{\xi}_{ik}$. Letting

$$\bar{\mathbf{T}}_{jk} \triangleq \bar{\mathbf{T}}_{ij}^{-1} \bar{\mathbf{T}}_{ik}, \tag{48}$$

gives us

$$\exp(\boldsymbol{\xi}_{jk}^{\wedge}) = \exp(\boldsymbol{\xi}_{ij}'^{\wedge}) \exp(\boldsymbol{\xi}_{ik}'^{\wedge}). \tag{49}$$

Expanding the BCH formula in a similar manner to (34) and taking the expectation as in (35) results in the following up to first order:

$$\begin{aligned} E[\boldsymbol{\xi}_{jk} \boldsymbol{\xi}_{jk}^{\top}] \approx & E[\boldsymbol{\xi}'_{ij} \boldsymbol{\xi}'^{\top}_{ij}] + E[\boldsymbol{\xi}'_{ik} \boldsymbol{\xi}'^{\top}_{ik}] + \\ & + E[\boldsymbol{\xi}'_{ij} \boldsymbol{\xi}'^{\top}_{ik}] + E[\boldsymbol{\xi}'_{ik} \boldsymbol{\xi}'^{\top}_{ij}]. \end{aligned} \tag{50}$$

Evaluating the expectations in (50) results in

$$\begin{aligned} \boldsymbol{\Sigma}_{jk} \approx & \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \boldsymbol{\Sigma}_{ij} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^{\top} + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \boldsymbol{\Sigma}_{ik} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^{\top} - \\ & + \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \boldsymbol{\Sigma}_{ij,ik} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^{\top} - \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}} \boldsymbol{\Sigma}_{ij,ik}^{\top} \mathrm{Ad}_{\bar{\mathbf{T}}_{ij}^{-1}}^{\top}. \end{aligned} \tag{51}$$

Thus, uncertainty can be propagated through the relative pose function via (48) and (51). A summary of the uncertainty propagation methods derived in the last three sections is provided in Fig. 3.

B. Ignoring Correlation Leads to Inconsistency

Ignoring correlation of the associated perturbation variables leads to under/over-estimation of uncertainty, depending on if the correlation is positive or negative and if the operation being performed is pose composition or relative pose estimation. Fig. 5 shows an example of this for the case of relative pose with positive correlation.

To create these plots, we generated $M = 10000$ sets of two uncertain poses $\mathbf{T}_{g1}^{m} = \exp(\boldsymbol{\xi}_{g1}^{m\wedge}) \bar{\mathbf{T}}_{g1}$ and $\mathbf{T}_{g2} = \exp(\boldsymbol{\xi}_{g2}^{m\wedge}) \bar{\mathbf{T}}_{g2}$, with mean values

$$\bar{\mathbf{T}}_{g1} = \begin{bmatrix} 0.707107 & -0.707107 & 0 & 3 \\ 0.707107 & 0.707107 & 0 & 3 \\ -0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{52}$$

and

$$\bar{\mathbf{T}}_{g2} = \begin{bmatrix} 0.707107 & -0.707107 & 0 & 4.5 \\ 0.707107 & 0.707107 & 0 & 4.5 \\ -0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{53}$$

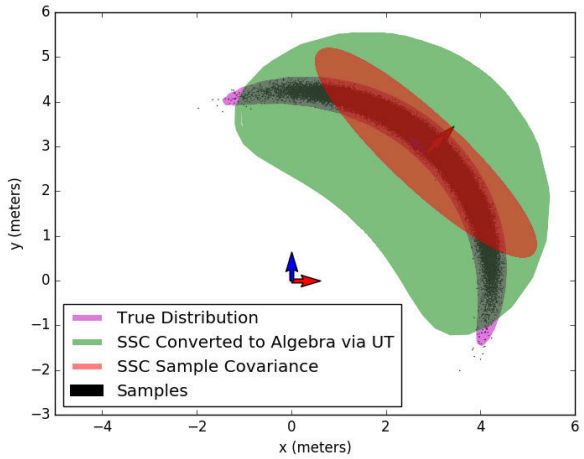


Fig. 6: The Unscented Transform (UT) [11] can be used to transform from an existing, coordinates based representation such as the one used in SSC (15) (shown in red) to the Lie algebra based representation (shown in green). However, the use of the SSC representation in the first place results in a loss of information and the resulting representation (shown in green) over approximates the true underlying distribution (shown in pink). All uncertainty bounds shown are 95% likely ellipsoids.

under the assumption that the perturbation variables ($\boldsymbol{\xi}_{g1}^{m}$) and ($\boldsymbol{\xi}_{g2}^{m}$) are jointly correlated in the Lie algebra with marginal covariance matrices

$$\begin{aligned}\boldsymbol{\Sigma}_{g1} &= \boldsymbol{\Sigma}_{g2} \\ &= \alpha \cdot \operatorname{diag}([0.005, 0.005, 1e-5, 1e-5, 1e-5, 0.006]),\end{aligned} \tag{54}$$

and cross covariance

$$\boldsymbol{\Sigma}_{g1,g2} = \alpha \cdot \operatorname{diag}([0.0005, 0.0005, 0, 0, 0, 0.005]), \tag{55}$$

where α is a scaling parameter. We then used the relative pose operation presented in this section to estimate $\mathbf{T}_{12}$ both with and without taking uncertainty into account.

Fig. 4 and Fig. 5 show that ignoring correlation leads to inconsistent estimates of uncertainty.

IX. Converting to a Lie algebra Based Representation

Although uncertainty characterization in the Lie algebra is more accurate than coordinate based methods, in some cases an existing estimation algorithm may output pose estimates in an alternative parameterization. This section describes how to convert from an existing parameterization such as the SSC multi-variate Gaussian representation (defined in (15) and (16)) to the proposed representation, as well as how to extract the mean and covariance of a jointly correlated set of poses from a MLE solution such as iSAM [34] or g2o [35] directly.

A. Converting from a Coordinate Based Representation

Assuming we are given a mean parameter vector $\hat{\mathbf{x}}$ and associated covariance $\hat{\boldsymbol{\Sigma}}$ as defined in (16), to apply our proposed propagation methods, we must first transform $\hat{\mathbf{x}}$ and $\hat{\boldsymbol{\Sigma}}$ into into our proposed representation such that the means of the associated transformations are represented in the group space via $\bar{\mathbf{T}}_{1:n}$ and the perturbations are represented as multivariate Guassian random variables, $\boldsymbol{\xi}_{1:n}$, with covariance $\boldsymbol{\Sigma}_{1:n}$ in the associated Lie algebra as defined in (28) and (29). Converting from a coordinates based method for uncertainty characterization to the Lie algebra representation is a non-linear operation. As such, the unscented transform [11] is a better choice for converting between representations than a first order linearization method.

The unscented transform, first proposed by Julier and Uhlmann [36] in 1997, takes an input Gaussian distribution defined in one space and a non-linear function that maps the input space to an output space and finds a Gaussian approximation for the transformed distribution. This is done by using the input mean and covariance to create a deterministic set $\mathcal{X}_{1:K}$ of $K = 2m+1$ sigma points, where m is the dimension of the input space. These sigma points are then passed through the non-linear function and a weighted mean and sample covariance is used to find a Gaussian approximation to the output distribution.

Assuming we have a function $f : \mathbb{R}^d \mapsto \mathcal{G}$ that maps from a parameter vector, $\mathbf{x}_i$, to an element of the Lie group, $\mathcal{G}$, we can define a function ℓ_i that takes a perturbed parameter vector $\tilde{\mathbf{x}}_i$, centers it on the identity, and transforms it to the Lie algebra space as follows:

$$\ell_i(\tilde{\mathbf{x}}_i) = \log(f(\tilde{\mathbf{x}}_i) \cdot f(\hat{\mathbf{x}}_i)^{-1}), \tag{56}$$

where $\hat{\mathbf{x}}_i$ is the mean parameter vector associated with $\tilde{\mathbf{x}}_i$.

We can then use the standard unscented transform [36, (12)] to generate a set $\mathcal{X}_{1:K}$ of $2nd+1$ sigma points from $\hat{\mathbf{x}}$ and $\hat{\boldsymbol{\Sigma}}$, where n is the number of modeled Lie group elements. After which, the predicted Lie group representation for the jointly correlated poses can be obtained as follows:

$$\bar{\mathbf{T}}_i = f(\hat{\mathbf{x}}_i) \tag{57}$$

$$\boldsymbol{\Sigma}_{1:n} = \sum_{k=1}^{K} \mathcal{W}_k \ell(\mathcal{X}_k) \ell(\mathcal{X}_k)^\top, \tag{58}$$

where ℓ is a vectorized version of ℓ_i and $\mathcal{W}_k$ is the standard weight from [36, (12)]. Fig. 6 shows a result of this process.

While, in many cases the use of the Unscented Transform may be the best that can be done, the use of the coordinates based representation (even as an intermediate representation) does lead to some loss of information. This loss of information can be avoided if we can directly represent the uncertainty in the proposed framework when extracting it from the prior estimation solution.

B. Extracting Pose Uncertainty from a MLE Solution

State-of-the-Art Pose Graph SLAM solvers find a solution by estimating the set of robot poses that maximizes the likelihood of the observed measurements [34, 37–40]. Traditional, iterative non-linear solvers [34, 35, 37, 38] do this by building up a measurement Jacobian, A, with columns that correspond to elements of the parameter vector $\mathbf{x}$ and with block rows that correspond to weighted, measurement residual error functions that minimize the error between predicted measurements and

what was actually measured. At each iteration, this measurement Jacobian is used to form a linear least squares problem of the following form:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|A\mathbf{x} - \mathbf{b}\|^2, \tag{59}$$

where $\mathbf{b}$ is the measurement vector not needed for the following derivation. The algorithm alternates between solving this linear least squares optimization problem and relinearizing A around $\hat{\mathbf{x}}$ until convergence [34, 37]. Algorithms such as those described in [39] and [40] that provide a guarantee of global optimality formulate the problem slightly differently, but once a solution has been obtained, a matrix A can still be formed by linearizing the cost around the current solution.

After a solution has been reached, an estimate of the uncertainty of that solution can be found by using A to form the information matrix $\mathcal{I} = A^\top A$ and using the non-zero elements of its Cholesky factorization $\mathcal{I} = R^\top R$ to calculate the necessary elements of the marginal covariance $\hat{\boldsymbol{\Sigma}}$ as detailed in [25]. The trick to extracting this covariance with respect to $\boldsymbol{\xi}_{1:n}$ as opposed to $\hat{\mathbf{x}}$ is to make sure that the Jacobian A used to form $\mathcal{I}$ is evaluated with respect to $\boldsymbol{\xi}_{1:n}$ as opposed to $\hat{\mathbf{x}}$. This can be done by numerically evaluating the Jacobian and by perturbing $\boldsymbol{\xi}_{1:n}$ around $\mathbf{0}$ and propagating that perturbation to the linearization point by means of the exponential function as opposed to perturbing the parameters of $\hat{\mathbf{x}}$ directly. Doing this enables the direct extraction of $\boldsymbol{\Sigma}_{1:n}$ and results in increased accuracy because $\boldsymbol{\xi}_{1:n}$ lies in a vector space, while $\hat{\mathbf{x}}$ does not.

X. Evaluation

This section evaluates the proposed uncertainty characterization method by performing two experiments. The first, performs a parameter sweep over the experiment introduced in Section VI-B. The second, extracts covariance information from the result of a Pose Graph SLAM algorithm and compares the predicted relative pose covariance with the sample covariance obtained from Monte Carlo.

A. Compounding Correlated Odometry

The exact accuracy of our proposed uncertainty characterization method is dependent on a variety of parameters including the number of poses compounded end-to-end and the rotation and translation noise. We explore this dependence by performing a parameter sweep across each of these parameters based on the experiment introduced in Section VI-B.

We began by varying the number of poses in the trajectory (N) and held both noise scale parameters fixed at $\sigma_t = \sigma_r = 3$. The results are shown in Fig. 7. As the number of steps increases, the accuracy of the final estimate drops off, this is because each of the methods are only characterizing uncertainty up-to first order and the lost higher order information builds up as the number of poses increases.

For the noise parameter experiments, we varied the noise scale parameters σ_r and σ_t in turn, while keeping the number of poses fixed at $N = 10$ and the non-varying noise parameter fixed at 3. The results are shown in Fig. 8. Rotation noise has a much more significant effect, again this is because of the lost

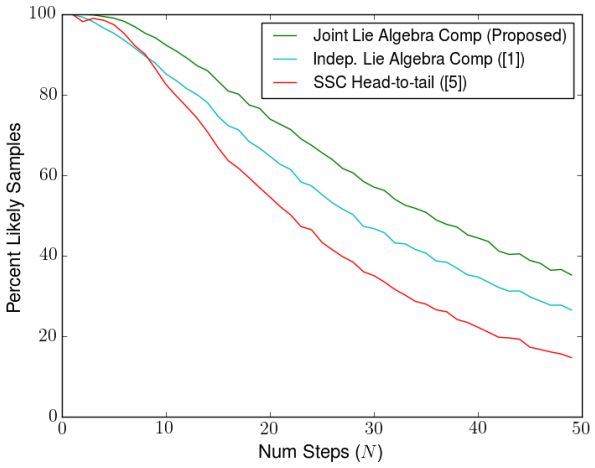


Fig. 7: The percentage of final samples that fell within the 99.9% likely covariance ellipsoid as a function of the number of poses in the trajectory sequence. All methods drop off as the number of poses is increased, however our proposed method is consistently most accurate.

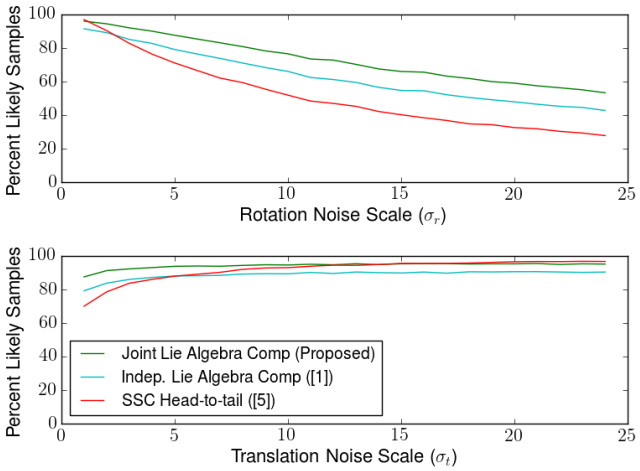
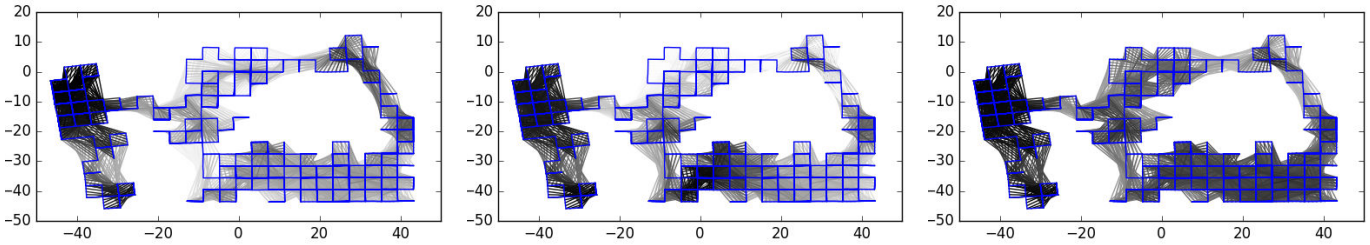


Fig. 8: The percentage of final samples that fell within the 99.9% likely covariance ellipsoid as a function of the rotation and translation noise. For the rotation noise sweep, translation noise was held constant at $\sigma_t = 3$ and for the translation noise sweep, rotation noise was held constant at $\sigma_r = 3$. Note that increases in rotation noise have the largest negative effect.

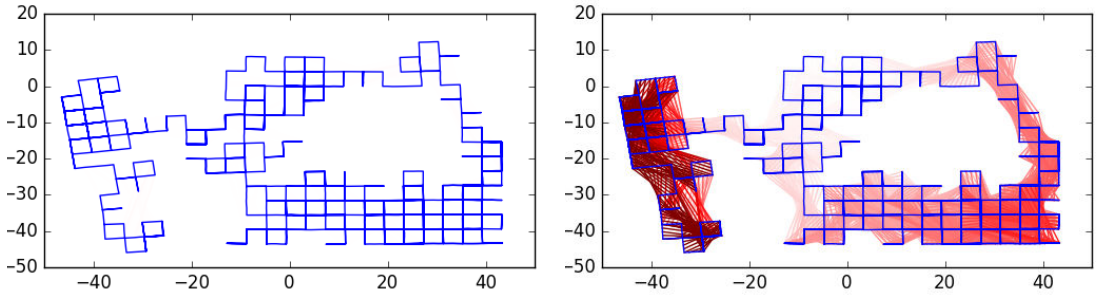
TABLE I: Summary of covariance error statistics for proposed Lie algebra relative pose estimation when correlation is taken into account and ignored for a total of 44425 pose pairs extracted from a solution to the Manhattan3500 dataset [41], with pose offsets ranging from 5 to 500 nodes.

Method	Covariance Error Mean	Covariance Error Std. Dev.
Proposed Lie algebra Rel. Pose	$0.00675104 \pm 2.2e^{-4}$	0.0461455
Proposed (Ignoring Correlation)	$2.05667 \pm 1.0e^{-2}$	2.12371

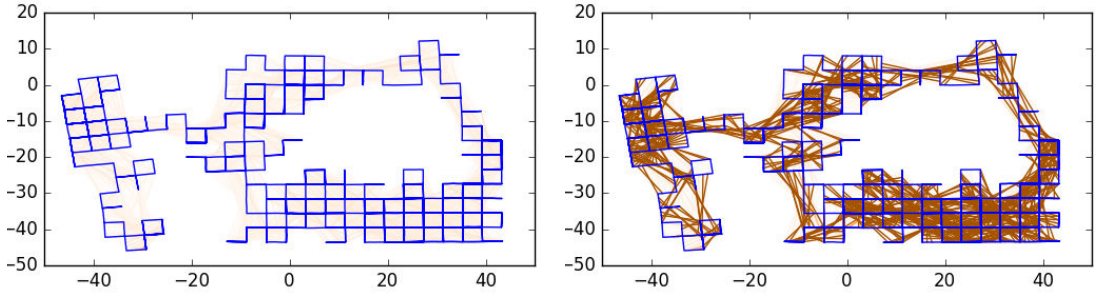
higher-order information (either in terms of the higher order terms of the BCH formula or through linearization for SSC). Both Lie group methods consistently outperform SSC except when rotation noise is very low and translation noise is very high. This is because when rotation error is very low, the non-linearity of the transformation becomes almost negligible and because the increased translation error causes the covariance ellipsoid to increase to the point where it includes the final robot position. In addition, the joint composition method in



(a) Pose Pair Correlation - X (b) Pose Pair Correlation - Y (c) Pose Pair Correlation - Heading



(d) Proposed Lie Algebra Relative Pose Cov. Error (e) Proposed Relative Pose Cov. Error (Ignoring Correlation)



(f) Proposed Lie Algebra Relative Pose Normalized Cov. Error (g) SSC Relative Pose Normalized Cov. Error

Fig. 9: A visualization of the correlation and covariance error for 3450 pose pairs with an offset of 50 nodes extracted from a solution of the Manhattan3500 dataset [41]. (a), (b), and (c) show lines between the two poses in the pair colored by the correlation coefficient corresponding to the degree of correlation between the two poses in the x, y, and θ dimensions respectively. White corresponds to a correlation coefficient of 0 and black to a coefficient of 1. (d) and (e) show the covariance error with respect to Monte Carlo for relative pose estimation in the Lie algebra when correlation is and is not taken into account. Dark red corresponds to a covariance error of at least 2 standard deviations above the mean (with respect to when correlation is ignored), while white corresponds to a covariance error of 0. (f) and (g) show the normalized covariance error with respect to Monte Carlo for the proposed method and SSC [5]. In this case, dark orange corresponds to a covariance error of at least 2 standard deviations above the mean (with respect to SSC relative pose extraction), while white corresponds to a covariance error of 0. Thus, the left and right columns should be compared and the darker the color of the plotted relative pose transformations, the larger the error.

TABLE II: Summary of normalized covariance error statistics for proposed and SSC [5] relative pose estimation for a total of 44425 pose pairs extracted from a solution to the Manhattan3500 dataset [41], with pose offsets ranging from 5 to 500 nodes.

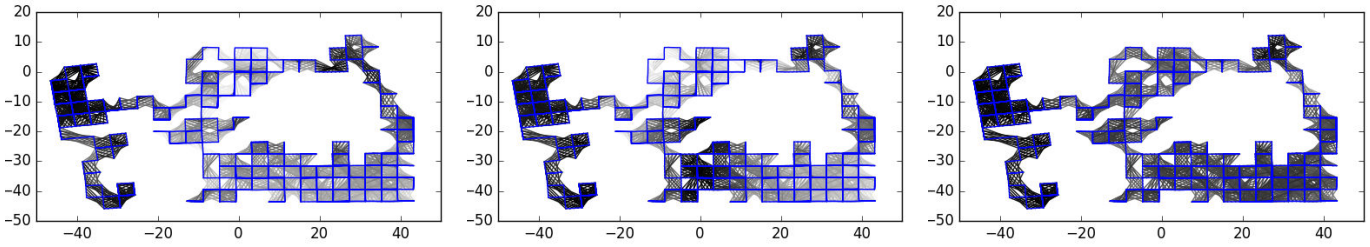
Method	Normalized Cov. Error Mean	Normalized Cov. Error Std. Dev.
Proposed Rel. Pose	$0.0493121 \pm 1.7e^{-4}$	0.0353072
SSC Tail-to-Tail	$0.28778 \pm 2.0e^{-3}$	0.411625

the Lie algebra is consistently more consistent than when correlation is ignored. For all three parameter sweeps, the induced correlation was held fixed at $\rho = 0.4$.

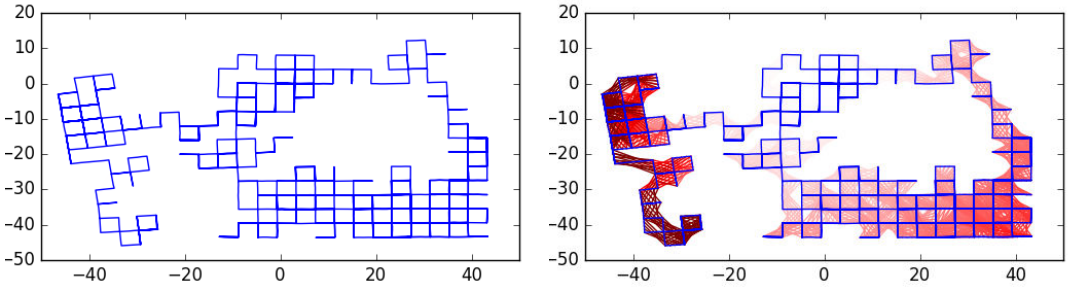
B. Extracting Relative Pose From a SLAM Solution

To evaluate uncertainty propagation through the relative pose operation, we used the Manhattan 3500 dataset [41] to generate a set of pose pairs on which to perform the relative pose operation. After using iSAM [34] to find a solution to the SLAM problem, we extracted joint mean and covariance as described in Section IX-B for pairs of poses at offsets varying from 5 to 50 in increments of 5 as well as for offsets of 100, 200, and 500 nodes. A visualization of the extracted pose pairs and the correlation between them for offsets of 50 are shown in Fig. 9 (a-c). For each pose pair, we plot a line between the two poses colored according to the degree of correlation between the two poses with respect to the x, y, and θ dimensions. Equivalent visualizations for offsets of 10, 100, 200, and 500 are shown in Fig. 10, Fig. 11, Fig. 12, and Fig. 13.

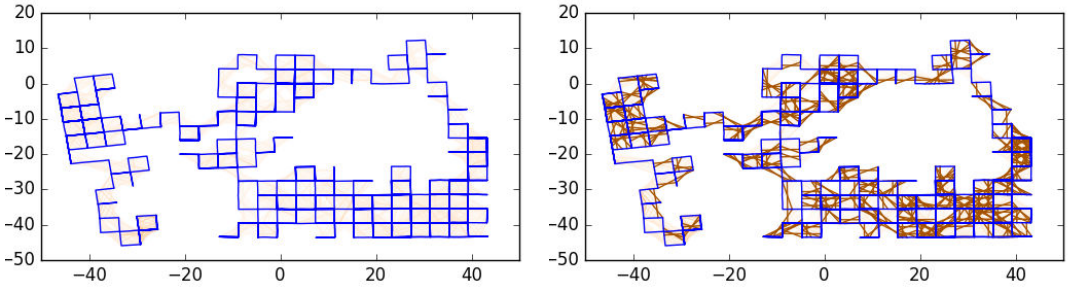
For each extracted pose pair, we performed a Monte Carlo simulation to determine a ground truth relative pose distribution against which we could compare the first-order



(a) Pose Pair Correlation - X (b) Pose Pair Correlation - Y (c) Pose Pair Correlation - Heading



(d) Proposed Lie Algebra Relative Pose Cov. Error (e) Proposed Relative Pose Cov. Error (Ignoring Correlation)



(f) Proposed Lie Algebra Relative Pose Normalized Cov. Error (g) SSC Relative Pose Normalized Cov. Error

Fig. 10: A visualization of the correlation and covariance error for 3490 pose pairs with an offset of 10 nodes extracted from a solution of the Manhattan3500 dataset [41]. The color schemes match those of Fig. 9.

propagation methods being investigated. First, the mean and joint marginal covariance for each pose pair was extracted from the iSAM solution. Then, for a given pair, a large number of sample pose pairs were generated from the extracted pose pair distribution and the relative pose between each of those pairs was evaluated. This gave us a sample distribution of the relative transformation between the two poses in the original pair which was used as groundtruth. We then repeated the process for each joint pose pair distribution extracted from the dataset.

Using this Monte Carlo derived relative pose covariance as a reference, we can use the following covariance error metric to evaluate the predicted covariance for the various methods we compare:

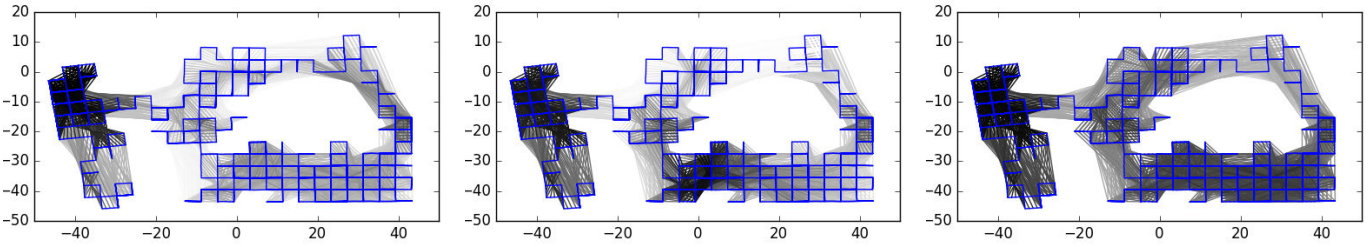
$$\epsilon \triangleq \sqrt{\mathrm{tr}\left((\boldsymbol{\Sigma} - \boldsymbol{\Sigma}_{mc})^{\top}(\boldsymbol{\Sigma} - \boldsymbol{\Sigma}_{mc})\right)}, \tag{60}$$

where $\boldsymbol{\Sigma}_{mc}$ is the sample relative pose covariance matrix generated via Monte Carlo for the given pose.

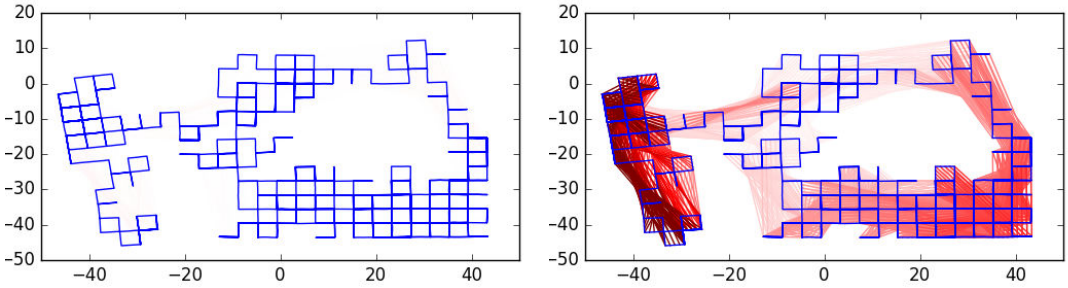
To evaluate our method proposed in (51) when taking into account correlation (using all four terms in (51)) or ignoring correlation (using only the first two terms in (51)), we used both methods to propagate pose uncertainty through the relative pose transformation for each pair and evaluated the error of the predicted covariance according to (60). Summary statistics are shown in Table I and visualizations of the error for offsets of 50, 10, 100, 200, and 500 are shown in (d-e) of Fig. 9, Fig. 10, Fig. 11, Fig. 12, and Fig. 13.

The results in Table I show that ignoring correlation can lead to a covariance error more than three orders of magnitude higher than if correlation is taken into account.

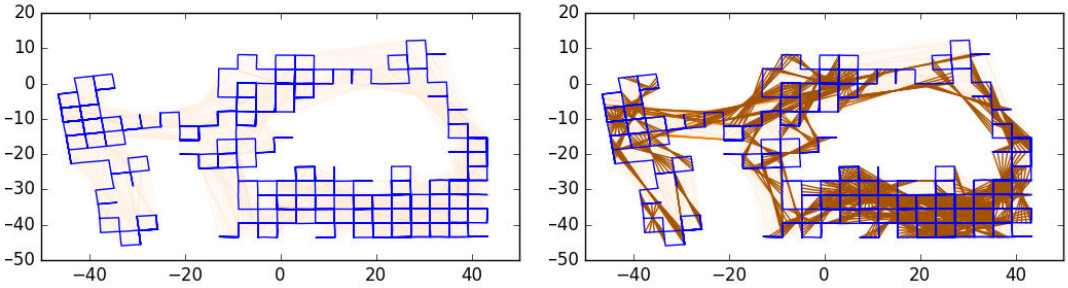
To compare our proposed method to SSC [5], we performed a similar experiment except that the Monte Carlo "groundtruth" covariance for SSC was derived by taking the sample relative poses from the previous experiment and converting them to the parameter vector format described in (15) and taking the sample covariance. To fairly compare our proposed method and SSC, the covariance matrices were normalized by the Frobenious norm of the Monte Carlo covariance matrix before evaluating the covariance error as defined in (60). Summary statistics of the experiment are shown in Table II and visualizations of the error for offsets of 50, 10, 100, 200, and 500 are shown in (f-g) of Fig. 9,



(a) Pose Pair Correlation - X (b) Pose Pair Correlation - Y (c) Pose Pair Correlation - Heading



(d) Proposed Lie Algebra Relative Pose Cov. Error (e) Proposed Relative Pose Cov. Error (Ignoring Correlation)



(f) Proposed Lie Algebra Relative Pose Normalized Cov. Error (g) SSC Relative Pose Normalized Cov. Error

Fig. 11: A visualization of the correlation and covariance error for 3400 pose pairs with an offset of 100 nodes extracted from a solution of the Manhattan3500 dataset [41]. The color schemes match those of Fig. 9.

Fig. 10, Fig. 11, Fig. 12, and Fig. 13.

It should be noted that this is not a perfect comparison because the Monte Carlo covariance to which SSC is being compared is a multivariate Gaussian fit to a set of parameter vectors that cannot be accurately modeled as a Gaussian. However, the results do show that even when modeling the true error in the format assumed by the SSC representation, our proposed Lie algebra based method results in an order of magnitude lower covariance error than SSC (see Table II).

C. Loop-closure Evaluation on a Real-World Dataset

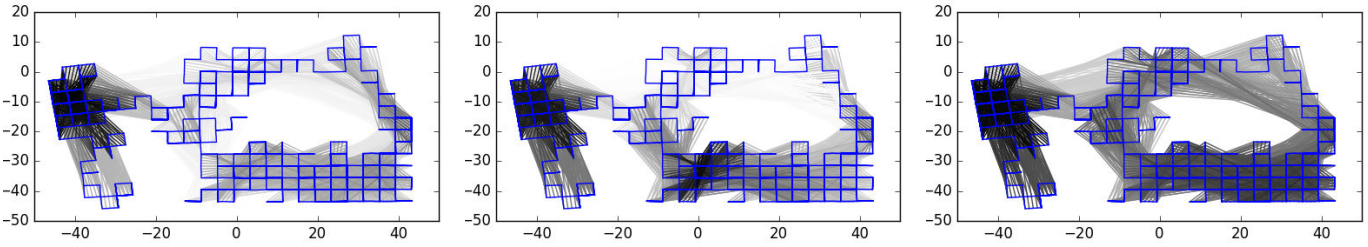
The results of the previous experiment show that our proposed method results in more accurate characterizations of pose uncertainty when used to evaluate the relative pose between pairs of poses extracted from an estimated SLAM solution. This increased accuracy is important when evaluating potential loop-closures. To emphasize this, we performed an additional experiment on the CSAIL real-world dataset collected at MIT [42].

We first divided the factors in the graph into a group of verified accurate loop-closures and odometry. We then used iSAM [34] to estimate a solution to the graph using odometry only and extracted the joint block covariance for the pairs of nodes associated with each loop-closure. As before, we extracted this joint covariance twice using both the proposed parameterization and that used by SSC. We then estimated the relative pose between the two base poses using the method proposed in Section VIII and via the SSC tail-to-tail operation and evaluated the proposed loop-closures by thresholding the squared Mahalanobis distance.

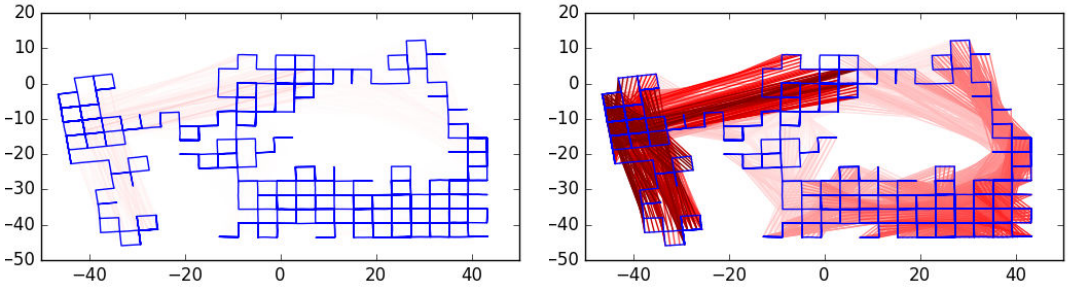
TABLE III: Percentage of true loop-closures who's Mahalanobis distance value fell below the chi-squared threshold for the proposed method and SSC.

Method	Loop-closures Accepted
Proposed	100.0 %
SSC	89.1 %

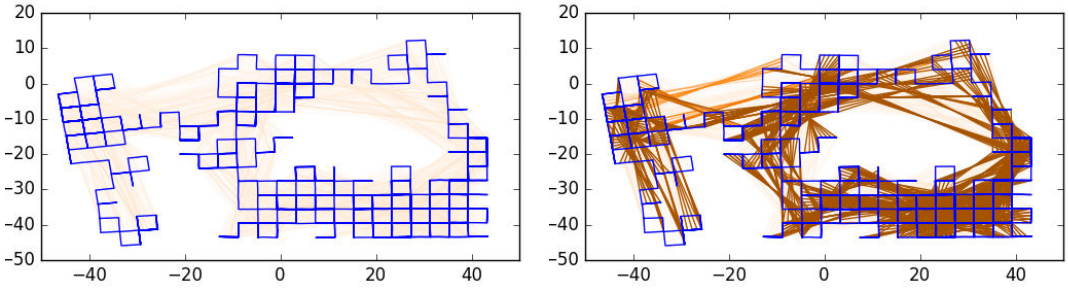
For a given loop-closure $\mathbf{T}_{jk}$, the SSC Mahalanobis distance was evaluated by applying the SSC tail-to-tail operation to the mean and associated joint covariance for the two poses $\mathbf{x}_{gj}$ and $\mathbf{x}_{gk}$ extracted from iSAM to obtain $\hat{\mathbf{x}}_{jk}^{odom}$ and $\mathbf{\Sigma}_{jk}^{odom}$ and



(a) Pose Pair Correlation - X (b) Pose Pair Correlation - Y (c) Pose Pair Correlation - Heading



(d) Proposed Lie Algebra Relative Pose Cov. Error (e) Proposed Relative Pose Cov. Error (Ignoring Correlation)



(f) Proposed Lie Algebra Relative Pose Normalized Cov. Error (g) SSC Relative Pose Normalized Cov. Error

Fig. 12: A visualization of the correlation and covariance error for 3300 pose pairs with an offset of 200 nodes extracted from a solution of the Manhattan3500 dataset [41]. The color schemes match those of Fig. 9.

then calculating the distance as follows:

$$\delta_{SSC} = (\hat{\mathbf{x}}_{jk}^{odom} - \mathbf{x}_{jk})^{\top} \boldsymbol{\Sigma}_{jk}^{odom^{-1}} (\hat{\mathbf{x}}_{jk}^{odom} - \mathbf{x}_{jk}), \qquad (61)$$

where $\mathbf{x}_{jk}$ is $\mathbf{T}_{jk}$ expressed as a parameter vector in $\mathbb{R}^3$ as defined in [5] for $\mathrm{SO}(2)$.

Under the proposed parameterization, the Mahalanobis distance calculation was performed by using the relative pose operation defined in Section VIII to estimate $\bar{\mathbf{T}}_{jk}^{odom}$ and $\boldsymbol{\Sigma}_{jk}^{odom}$ from the poses $\mathbf{T}_{gj}$ and $\mathbf{T}_{gk}$ and their joint covariance matrix extracted from iSAM and then calculating the distance as follows:

$$\delta_{Lie} = \log(\mathbf{T}_{jk} \bar{\mathbf{T}}_{jk}^{odom^{-1}})^{\top} \boldsymbol{\Sigma}_{jk}^{odom^{-1}} \log(\mathbf{T}_{jk} \bar{\mathbf{T}}_{jk}^{odom^{-1}}). \qquad (62)$$

Finally, both distances where compared against the chi-squared threshold value 7.814 corresponding to a 95 percent likelihood for three degrees of freedom. The results are shown in Fig. 14, Fig. 15, and Table III.

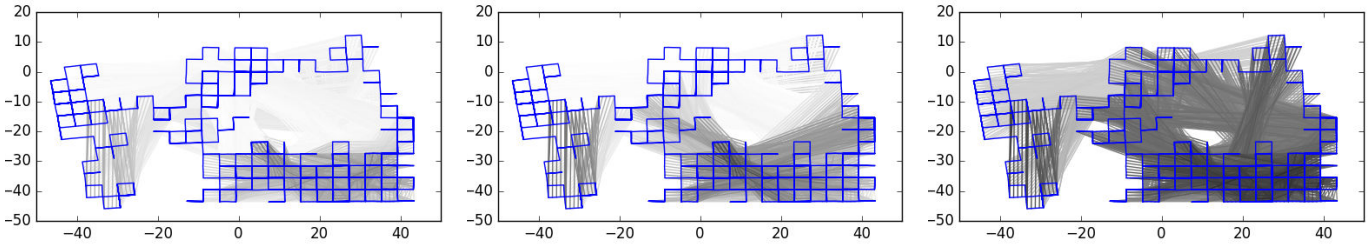
When using the proposed Lie algebra based method, 100 percent of the verified loop-closures fall within the expected chi-squared bounds, while when using SSC more than ten percent fall outside it (see Fig. 15 and Table III).

XI. Conclusion

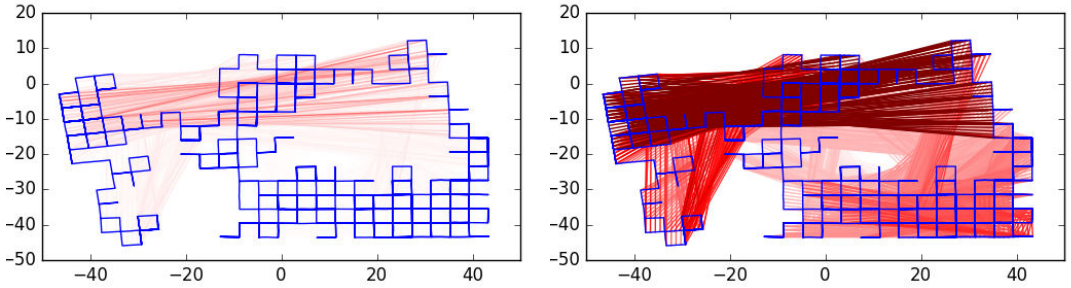
Recent interest has shown that pose uncertainty characterization through the use of the Lie algebra leads to more accurate uncertainty propagation. However, recent methods that represent uncertainty via a perturbation defined in the Lie algebra assume that individual poses are independent from one another. This is not always the case, for example, after solving the SLAM problem the resulting poss are densely correlated. In addition, existing publications have primarily focused on pose composition, ignoring the equally important inverse and relative pose operations.

This paper describes how to represent multiple jointly correlated poses while using the Lie algebra to characterize uncertainty. We also derive the equivalent of the Smith, Self, and Cheeseman [5] pose composition, pose inverse, and relative pose operations when using the proposed framework.

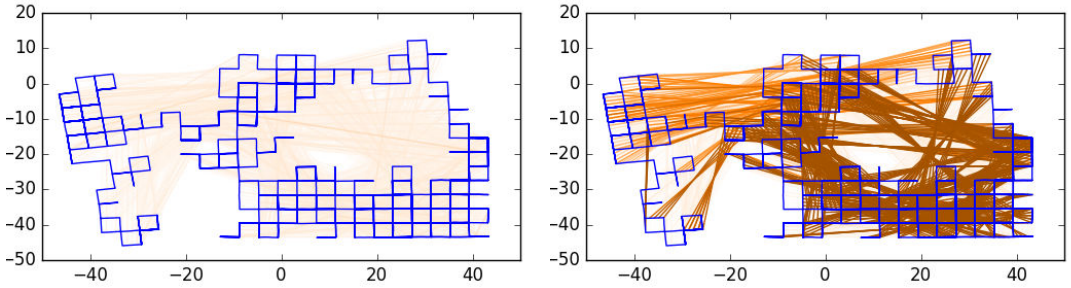
Our results show that the proposed methods increase the accuracy of data association consistency checks when extract-



(a) Pose Pair Correlation - X (b) Pose Pair Correlation - Y (c) Pose Pair Correlation - Heading



(d) Proposed Lie Algebra Relative Pose Cov. Error (e) Proposed Relative Pose Cov. Error (Ignoring Correlation)



(f) Proposed Lie Algebra Relative Pose Normalized Cov. Error (g) SSC Relative Pose Normalized Cov. Error

Fig. 13: A visualization of the correlation and covariance error for 3000 pose pairs with an offset of 500 nodes extracted from a solution of the Manhattan3500 dataset [41]. The color schemes match those of Fig. 9.

ing pose uncertainty information from a pose graph SLAM solution for loop-closure selection [10]. It can also be used to compose odometry measurements that are potentially correlated with one another such as is the case in the presence of wheel slip due to wheel irregularity or loose gravel in the environment. By accurately modeling pose uncertainty, the proposed method increases the robustness and reliability of autonomous navigation. Finally, we have released an open source C++ library implementation of our method as described in the Appendix.

APPENDIX

We have released an open source C++ library implementation of our method. It can be downloaded at: https://bitbucket.org/jmangelson/lie. We have tried to design it to be simple, intuitive, and easily extendable. This appendix contains an introduction to our C++ library implementation.

A. *Creating Known and Uncertain* SE(3) *Objects*

Creating both known and uncertain SE(3) objects is syntactically easy. After importing the library a variety of constructors can be used to create known SE(3) transformations:

```
#include <lie/se3.hpp>

// Via rotation and translation
Eigen::MatrixXd R(3,3); // 3x3 matrix
Eigen::VectorXd t(3); // 3x1 vector
Lie::SE3 T_12(R, t);

// Via translation and Euler angle params
Lie::SE3 T_23(x, y, z, theta, phi, psi);
```

Independent and jointly distributed sets of unknown poses can also be created by passing in a mean value or a tuple of mean values and a covariance matrix.

```
Lie::SE3 T_ab_mu(R_ab, t_ab);
Eigen::MatrixXd Sigma_ab(6,6);
auto T_ab_uncertain =
  Lie::make_uncertain_state(
    T_ab_mu, Sigma_ab);

Lie::SE3 T_ac_mu(R_ac, t_ac);
```

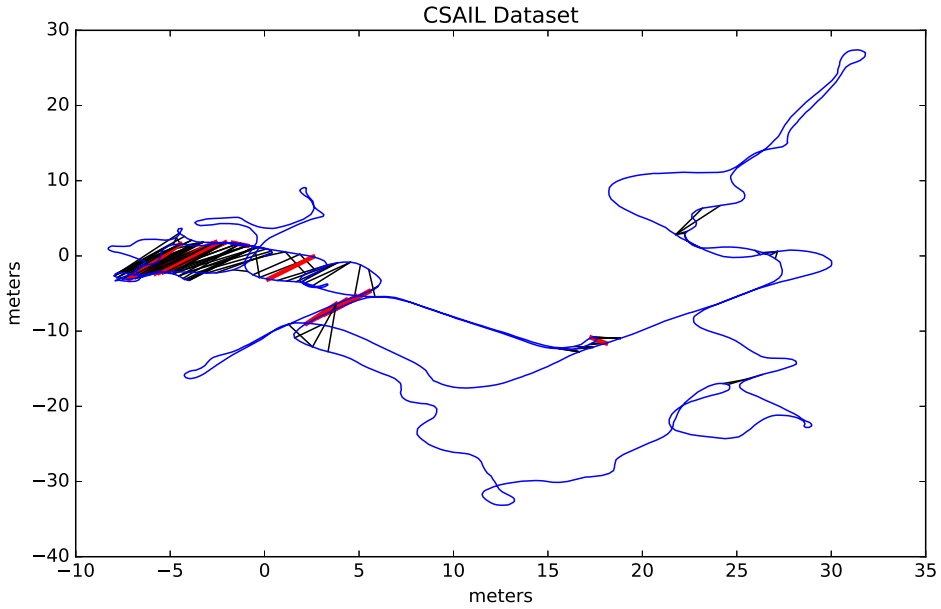


Fig. 14: A visualization of the odometry of the CSAIL dataset [42] along with the loop-closures passing the Mahalanobis distance threshold test for both the proposed method and SSC. The odometry trajectory is shown in blue. Loop-closures accepted by both methods are shown in black, while those not accepted by SSC and only accepted by the proposed method are shown in red.

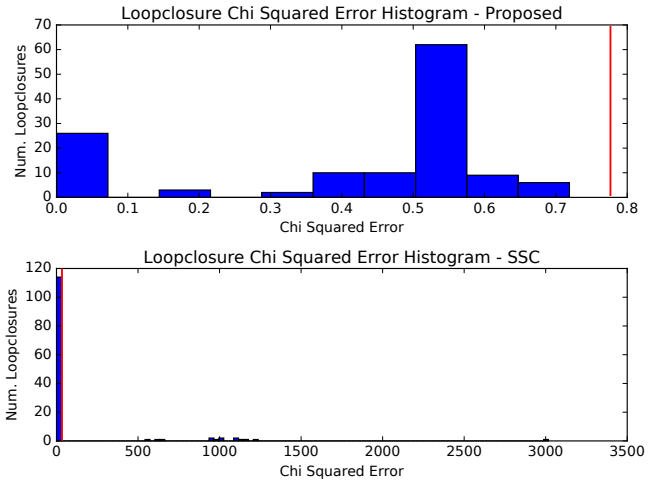


Fig. 15: Histograms of the Mahalanobis distance values for each true loop-closure. For three dimensional Chi-squared random variables, 95% of these values should fall below a threshold of 7.814 (shown as a red vertical bar). Using the proposed method 100% fall below the threshold, while many true loop-closures fall above the threshold when using SSC.

```
Eigen::MatrixXd Sigma_Joint(12,12);
auto refs =
  Lie::make_uncertain_state(
    std::make_tuple(T_ab_mu, T_ac_mu),
    Sigma_Joint);
auto T_ab_uncertain = std::get<0>(refs);
auto T_ac_uncertain = std::get<1>(refs);
```

The `Lie::make_uncertain_state` function returns either a single $\mathrm{SE}(3)$ reference object or a tuple of such objects that can then be used to perform operations.

B. Performing Operations

The pose composition, inverse, and relative pose operations can be applied interchangeably regardless of whether or not individual $\mathrm{SE}(3)$ objects are known or uncertain via the `Lie::compose`, `inverse`, and `Lie::between` functions respectively. The compiler automatically determines whether or not the individual objects are known/unknown or independent/jointly distributed and apply the appropriate formulation.

```
// Composing known poses
auto T_13 = Lie::compose(T_12, T_23);

// Invert both known and uncertain poses
auto T_21 = T_12.inverse();
auto T_ba_uncertain =
  T_ab_uncertain.inverse();

// Calculating relative pose
auto T_bc_uncertain = Lie::between(
  T_ab_uncertain, T_ac_uncertain);
auto T_bc_uncertain2 = Lie::between(
  T_ab_known, T_ac_uncertain);
```

Each function returns a new (from then on assumed independent) $\mathrm{SE}(3)$ reference object that can be used for additional operations as necessary.

C. Additional Lie groups and Alternative Conventions

In addition, because the uncertainty propagation method we present simply uses the properties of Lie groups, it can easily be applied to other Lie group types. We have also implemented the `Lie::SO3`, `Lie::SE2`, and `Lie::SO2` classes for the $\mathrm{SO}(3)$, $\mathrm{SE}(2)$, and $\mathrm{SO}(2)$ Lie groups and plan to extend it to other Lie group types as needed.

Finally, we also plan to extend the library to include alternative conventions. For example, although in this paper we follow [1] and use left handed multiplication of the exponential to define random variables as in (14), there also situations where it makes sense to use multiplicaiton on the right hand side [29]. Additionally, the convention in this paper defines $\boldsymbol{\xi} := [\boldsymbol{\rho}^\top \boldsymbol{\phi}^\top]^\top$ as in (6), while it is also commonly defined as $\boldsymbol{\xi} := [\boldsymbol{\phi}^\top \boldsymbol{\rho}^\top]^\top$. For example, GTSAM [43] uses the latter convention for $\boldsymbol{\xi}$ and exports covariances using right handed exponential multiplication by default. We plan to extend our library to directly implement these alternative conventions.

REFERENCES

[1] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Trans. on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[2] M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, 2001.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[4] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, 1986.

[5] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Auton. Robot.*, pp. 167–193, 1990.

[6] Y. Wang and G. S. Chirikjian, "Nonparametric second-order theory of error propagation on motion groups," *The International journal of robotics research*, vol. 27, no. 11-12, pp. 1258–1273, 2008.

[7] A. W. Long, K. C. Wolfe, M. J. Mashner, and G. S. Chirikjian, "The banana distribution is gaussian: A localization study with exponential coordinates," *Proc. Robot.: Sci. & Syst. Conf.*, vol. 265, Jul 2012.

[8] Y. Kim and A. Kim, "On the uncertainty propagation: Why uncertainty on lie groups preserves monotonicity?" in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Vancouver, BC, Canada, Sep. 2017, pp. 3425–3432.

[9] N. Sünderhauf and P. Protzel, "Towards a robust back-end for pose graph SLAM," in *Proc. IEEE Int. Conf. Robot. and Automation*, St. Paul, Minnesota, USA, May 2012, pp. 1254–1261.

[10] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *Proc. IEEE Int. Conf. Robot. and Automation*, Brisbane, Australia, May 2018, pp. 1–8.

[11] S. J. Julier, "The scaled unscented transformation," in *Proc. Amer. Control Conf.*, vol. 6, Anchorage, Alaska, USA, May 2002, pp. 4555–4559.

[12] D. J. MacKay and D. J. Mac Kay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[13] M. L. Rodríguez-Arévalo, J. Neira, and J. A. Castellanos, "On the importance of uncertainty representation in active SLAM," *IEEE Trans. on Robotics*, vol. 34, no. 3, pp. 829–834, 2018.

[14] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended kalman filter based SLAM," *IEEE Trans. on Robotics*, vol. 23, no. 5, pp. 1036–1049, 2007.

[15] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New York, 2005.

[16] R. M. Murray, Z. Li, and S. Shankar Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

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

[18] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 1: Classical Results and Geometric Methods*. Springer Science & Business Media, 2009.

[19] ——, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011, vol. 2.

[20] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME—J. Basic Eng.*, pp. 82:35–45, 1960.

[21] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. on Robotics*, vol. 7, no. 3, pp. 376–382, 1991.

[22] P. Moutarlier and R. Chatila, "Stochastic multisensory data fusion for mobile robot location and environment modeling," in *Proc. Int. Symp. Robot. Res.*, Tokyo, Japan, Aug 1989.

[23] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. on Robotics*, vol. 17, no. 6, pp. 890–897, 2001.

[24] A. Barrau and S. Bonnabel, "Invariant kalman filtering," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 237–257, 2018.

[25] M. Kaess and F. Dellaert, "Covariance recovery from a square root information matrix for data association," *Journal of Robotics and Autonomous Systems, RAS*, vol. 57, no. 12, pp. 1198–1210, December 2009.

[26] S. Agarwal and K. Mierle, *Ceres Solver: Tutorial & Reference*, Google Inc.

[27] K. Tapp, *Matrix groups for undergraduates*. American Mathematical Soc., 2016, vol. 79.

[28] A. Baker, *Matrix groups: An introduction to Lie group theory*. Springer Science & Business Media, 2012.

[29] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *arXiv preprint arXiv:1904.09251*, 2019.

[30] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[31] R. Hartley, J. G. Mangelson, L. Gan, M. G. Jadidi, J. M. Walls, R. M. Eustice, and J. W. Grizzle, "Legged robot state-estimation through combined forward kinematic and preintegrated contact factors," in *Proc. IEEE Int. Conf. Robot. and Automation*, Brisbane, Australia, May 2018, pp. 1–8.

[32] D. O. Wheeler, D. P. Koch, J. S. Jackson, T. W. McLain, and R. W. Beard, "Relative navigation: A keyframe-based approach for observable gps-degraded navigation," *IEEE Control Syst. Mag.*, vol. 38, no. 4, pp. 30–48, 2018.

[33] E. Eade, "Lie groups for 2D and 3D transformations," accessed: 2018-12-05. [Online]. Available: http://ethaneade.com/lie.pdf

[34] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[35] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. and Automation*, Shanghai, China, May 2011, pp. 3607–3613.

[36] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068. International Society for Optics and Photonics, 1997, pp. 182–194.

[37] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1181–1203, 2006.

[38] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, 2012.

[39] D. Rosen, L. Carlone, A. Bandeira, and J. Leonard, "SE-Sync: A certifiably correct algorithm for synchronization over the Special Euclidean group," in *Proc. Int. Work. Algorithmic Foundations of Robot.*, 2016.

[40] J. G. Mangelson, J. Liu, R. M. Eustice, and R. Vasudevan, "Guaranteed globally optimal planar pose graph and landmark slam via sparse-bounded sum-of-squares programming," in *Proc. IEEE Int. Conf. Robot. and Automation*, Montreal, Canada, May 2019, pp. 1–8.

[41] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proc. IEEE Int. Conf. Robot. and Automation*, Orlando, Florida, May 2006, pp. 2262–2269.

[42] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, "Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization," in *Proc. IEEE Int. Conf. Robot. and Automation*, Seatle, Washington, USA, May 2015, pp. 4597–4604.

[43] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.

Joshua Mangelson received the B.S. degree in electrical engineering from Brigham Young University, Provo, UT, USA in 2014 and both M.S. and Ph.D. degrees in robotics from the University of Michigan, Ann Arbor, MI, USA in 2016 and 2019 respectively. He is currently a Post-Doctoral Researcher in the Field Robotics Center at Carnegie Mellon University and will start as an Assistant Professor in electrical and computer engineering at Brigham Young University in August 2020. His research interests lie in the development of navigation, mapping, perception, and planning algorithms with mathematical and performance guarantees that enable the design of reliable field robotic systems for operation in unstructured environments. He is especially interested in the development of large-scale multi-agent teams for autonomous inspection of underwater structures. He is the recipient of the IEEE ICRA Best Multi-Robot Paper Award and the IEEE OCEANS Best Poster Award both in 2018.

Maani Ghaffari received the Ph.D. degree from the Centre for Autonomous Systems (CAS), University of Technology Sydney, NSW, Australia, in 2017. He is currently an Assistant Research Scientist at the Robotics Institute and Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI, USA. His research interests include applied mathematics, robotic perception, machine learning, and planning under uncertainty with applications in robotics and autonomous systems.

Ram Vasudevan received the B.S. degree in electrical engineering and computer sciences, the M.S. degree in electrical engineering, and the Ph.D. degree in electrical engineering from the University of California at Berkeley in 2006, 2009, and 2012, respectively. He is currently an Assistant Professor in mechanical engineering with the University of Michigan, Ann Arbor, with an appointment in the University of Michigan's Robotics Program. His research interests include the development and application of optimization and systems theory to quantify and improve human and robot interaction.

Ryan M. Eustice received the B.S. degree in mechanical engineering from Michigan State University, East Lansing, MI, USA, in 1998 and the Ph.D. degree in ocean engineering from the Joint Program between the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA and the Woods Hole Oceanographic Institution (WHOI), Woods Hole, MA, USA, in 2005. Currently, he is the Senior Vice President of Automated Driving at the Toyota Research Institute and a Professor with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI, USA, with joint appointments in the Department of Electrical Engineering and Computer Science and in the Department of Mechanical Engineering. His research interests include autonomous navigation and mapping, automated driving, mobile robotics, and autonomous underwater vehicles.