On Sensor Pose Parameterization for Inertial Aided Visual SLAM

Markus Kleinert
Fraunhofer IOSB
Ettlingen, Germany
Email: markus.kleinert@iosb.fraunhofer.de

Uwe Stilla
Technische Universität München
München, Germany
Email: stilla@bv.tum.de

***Abstract*—When appropriate infrastructure is not available, localization of pedestrians becomes a difficult task. This is especially the case in urban or indoor scenarios, where satellite navigation is hindered due to occlusions or multipath effects. A promising alternative is to combine a small, low-cost IMU with a camera in order to exploit the complementary error characteristics of these devices by simultaneously estimating the positions of observed landmarks and the trajectory of the sensor system with a stochastic filter.**

In this work, a standard approach to parameterize the error in position and attitude estimates that is commonly used in GNSS-INS integration is compared to alternative parameterizations that are based on the twist representation of rigid body motions, which has gained increasing popularity in the literature. For this purpose, the error-state transition and measurement equations are formulated for the twist representation as well as for the standard approach. Finally, the different approaches are compared on a simulated and a real indoor dataset by applying an extended Kalman filter (EKF).

I. Introduction

A. Pedestrian navigation and visual SLAM

When operating in extended indoor environments, first responders strongly depend on their ability to reliably localize themselves in order to successfully accomplish their task. This is especially true when a team of first responders that is scattered over the environment has to perform a concerted action. In such a situation, a system that improves the situational awareness by displaying the locations of team members on a map would be very helpful.

Unfortunately, satellite navigation signals are usually not available or severely disturbed in urban scenarios due to occlusions or multipath effects. This motivates the research in alternative localization systems like Honeywell's GLANSER system [1], where utra-wideband radio signals are used to measure the distance to stationary anchor nodes, which have to be installed in advance at the site of operation.

Systems that do not depend on such external infrastructure are inevitably subject to an increasing error because they essentially perform dead reckoning, i.e., each position estimate depends on previous position estimates. An inertial measurement unit (IMU) is an example of a dead reckoning system because it measures acceleration and angular velocities which yield an estimate of the displacement of a sensor system when integrated. Since IMUs which offer the accuracy that is needed to obtain position estimates for extended periods of time with reasonable accuracy are both expensive and bulky, it seems promising to consider a combination of a low-cost IMU with a camera which compensates for the rapidly accumulating errors in the position estimate obtained by integrating the IMU's measurements by observing stationary landmarks in the video images. The different kinds of measurements obtained thereby are usually fused by applying a stochastic state estimator in order to estimate the trajectory of the combined camera-IMU system and the locations of the observed landmarks simultaneously, a problem that is commonly referred to as the simultaneous localization and mapping (SLAM) problem.

B. Contribution

This work compares different approaches to parameterize the sensor's attitude and position in an inertial aided visual SLAM system using an extended Kalman filter to estimate the SLAM state. More specifically, a parameterization that is commonly used for GNSS-INS integration which treats position and attitude errors separately is compared to alternative parameterizations where position and attitude errors are subsumed in a screw motion parameterized by an error twist. For this purpose, it is shown how the time prediction and covariance propagation equations for the error twist may be formulated. The observations reported in this work also pertain to alternative estimation techniques which also rely on a linearization of the motion and measurement models.

II. Rigid body motions

This section reviews the basic concepts that are needed for the description of rigid body motions. Most of the material presented here can also be found in [2], [3].

In this work the term *pose* refers to the position and attitude of a rigid body. The pose of a rigid body is commonly specified in terms of a coordinate transformation that converts the coordinates of points given in a coordinate system that is attached to the body into the coordinates of a fix reference coordinate system. Mathematically, these transformations form a group: The special Euclidean group SE(3) whose elements are the isometries of three-dimensional Euclidean space. More specifically, SE(3) is the semi-direct product of the special orthogonal group SO(3), whose elements represent rotations about the origin in three-dimensional space, and the three-dimensional vector space $\mathbb{R}^3$. The group operation is given

by the concatenation of coordinate transformations and the inverse of a group element is given by the inverse coordinate transformation. In practice, the elements of SE(3) are often given as 4×4 homogeneous matrices

$$T_a^b \quad = \quad \begin{bmatrix} C_a^b & {}^b\mathbf{p}_a \\ 0 & 1 \end{bmatrix}. \tag{1}$$

where $C_a^b \in \mathrm{SO}(3)$ is a rotation matrix and ${}^b\mathbf{p}_a \in \mathbb{R}^3$ is the position of frame a in the coordinates of frame b. However, the parameterization with homogeneous matrices is an overparameterization since the pose of a rigid body can be specified with six parameters. In addition, care has to be taken to retain the orthogonality of the rotation matrix C_a^b. Therefore, alternative parameterizations for the group elements of SE(3) are often suggested, especially in least-squares problems or filtering applications.

1) Twists: Fig. 1 illustrates two ways to describe the displacement of a coordinate system that is attached to a rigid body, which is moving with angular velocity ${}^b\boldsymbol{\omega}$ and velocity ${}^b\mathbf{v}$, both given in the body's coordinate system. While in Fig. 1a the displacement is interpreted as a translation of the origin along a straight line and a separate rotation, the same displacement may be described by a screw motion which consists of a displacement d along a screw axis and a rotation about that same axis as shown in 1b. According to Chasles's theorem, any proper rigid body motion that is not a pure translation can be described by such a screw motion, cf. [2]. This is the basis for the twist or screw representation of rigid body motions in terms of six-parameter vectors

$$\mathbf{t} \quad = \quad \left[\boldsymbol{\omega}^{\mathrm{T}}\, \boldsymbol{\xi}^{\mathrm{T}}\right]^{\mathrm{T}} \tag{2}$$

where $\boldsymbol{\omega}$ is the screw axis and $\boldsymbol{\xi}$ is another three-component vector that encodes the position of the screw axis and the amount of displacement along the screw axis. The norm of the screw axis is just the corresponding angle of rotation. In case of a pure translation, the screw axis vanishes and $\boldsymbol{\xi}$ is replaced by the translation vector.

2) Lie groups and Lie algebras: The importance of the twist representation becomes apparent when the geometric properties of the matrix group SE(3) are taken into account. The group elements $T \in \mathrm{SE}(3)$ form a continuous, differentiable manifold with the group operation and the inversion of group elements acting as differentiable mappings on the manifold. Such groups are also called Lie groups. For any path $T(s)$ in the Lie group that passes through the identity, i.e., $T(0) = I$, the derivatives at the identity $d/dsT(s)|_{s=0}$ are matrices that belong to a vector space, i.e., the tangent space at the identity element. The matrix commutator of two elements X, Y yields a binary operation on the tangent space:

$$[X, Y] \quad = \quad X \cdot Y - Y \cdot X \tag{3}$$

This operation is called Lie bracket and the tangent vector space augmented with this operation forms a Lie algebra. The Lie bracket is linear in both of its arguments. Thus, fixing

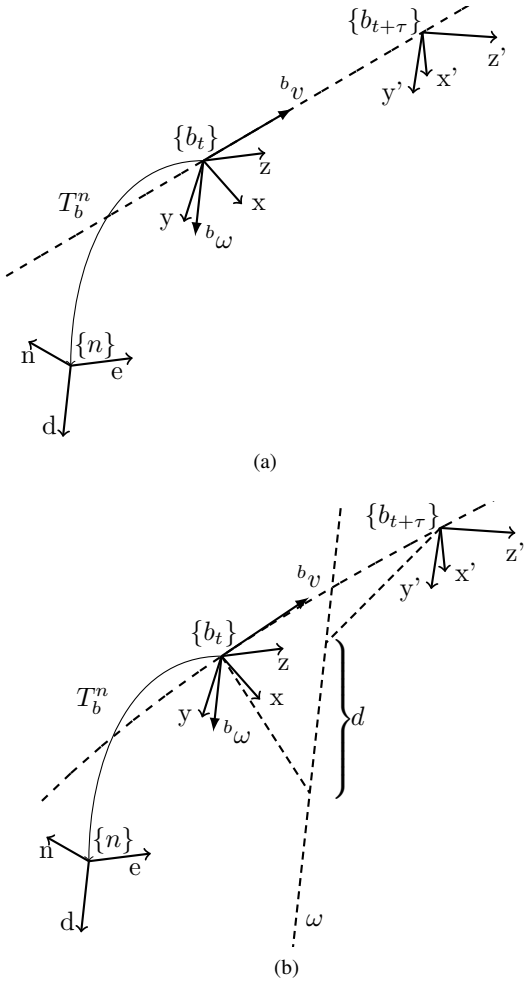

Fig. 1. Two ways to describe the displacement of a rigid body moving with velocity ${}^b\mathbf{v}$ and angular velocity ${}^b\boldsymbol{\omega}$. T_b^n denotes the coordinate transformation from the body coordinate system to the navigation frame. (a) Separate translation and rotation (b) Screw motion with displacement d and angle $\left\|{}^b\boldsymbol{\omega}\right\|$ about the screw axis $\boldsymbol{\omega}$.

the first argument yields a linear map on the Lie algebra, the adjoint representation of the algebra:

$$\mathrm{ad}_X(Y) \quad = \quad [X, Y] \tag{4}$$

In case of the group SE(3) its algebra is written as se(3) and its elements are given by the twists that describe the screw motion. The twists can be identified with the matrices that form the tangent space of SE(3) at the identity by the following mapping:

$$S_{\boldsymbol{\omega},\boldsymbol{\xi}} \quad = \quad \begin{bmatrix} \lfloor \boldsymbol{\omega} \rfloor_\times & \boldsymbol{\xi} \\ 0 & 0 \end{bmatrix} \tag{5}$$

Where $\lfloor \mathbf{v} \rfloor_\times$ is the skew-symmetric cross-product matrix associated with a vector $\mathbf{v}$.

3) The exponential map: The elements of the Lie algebra can be mapped to the elements of its corresponding Lie group by means of the matrix exponential:

$$T(S) \quad = \quad \exp(S) \quad = \quad \textstyle\sum_{k=0}^{\infty} \frac{S^k}{k!} \tag{6}$$

For a constant element $S \in \mathrm{se}(3)$ and a parameter $t \in \mathbb{R}$ the exponential map defines a curve

$$T(t) \quad = \quad \exp(St) \tag{7}$$

on SE(3) whose elements form a one-parameter subgroup. These curves are the solutions of the differential equation

$$\frac{dT(t)}{dt} = T(t) \cdot S \tag{8}$$

and S can be interpreted as the body-velocity if t is the time. On compact Lie groups the curves given by (7) have an additional desirable property: They are the unique geodesics through the identity and thereby the shortest paths between the identity element $T(0)$ and $T(t)$ on the Lie group w.r.t. a given Riemannian metric. However, as pointed out in [4] this is not true for general non-compact Lie groups like SE(3) where the notion of distance and shortest paths is complicated by the lack of a bi-invariant Riemannian metric.

4) Lie algebra in optimization problems: The properties presented in the previous sections motivate the parameterization of Lie group elements in terms of vectors in the corresponding Lie algebra in filtering and optimization problems. The general approach is as follows: In order to estimate the true value M of a Lie group element, one starts with an initial estimate $\hat{M}$ that is related to the true value by a multiplicative error $M = \hat{M} \cdot \exp(S)$ or $M = \exp(S) \cdot \hat{M}$. Then some objective function that depends on the Lie algebra element S is optimized. The resulting estimate $\hat{S}$ is finally used to correct $\hat{M}$ according to the multiplicative error function. Thereby it is possible to work with elements of a vector space (the Lie algebra), which can be added and multiplied by scalars in the usual way, while still preserving the geometric properties of the underlying manifold. In this way, the Lie algebra provides local coordinates for the Lie group nearby the estimated state.

III. Related work

A number of different approaches to represent rigid body motions can be found in the computer graphics and robotics literature. For instance, Dorst suggests the use of the conformal model of geometric algebra in order to represent Euclidean motions [5]. In contrast to homogeneous coordinates, which require different representations of Euclidean motions to move, e.g., points and lines, this allows for a unified treatment of motions for different geometric objects. The use of screw motions is suggested to interpolate between rigid body motions, but it is noted that no interpolation scheme can cope with changes in the body frame and the world frame simultaneously. As an explanation, it is stated that no bi-invariant metric exists on SE(3) that is invariant under moving world as well as moving body frame, as stated in [4].

However, despite the absence of a bi-invariant metric on general non-compact Lie groups like SE(3) it was shown by Mahony et al. that the Newton algorithm displays quadratic convergence when formulated in terms of the local coordinates provided by the Lie algebra [6].

Wu et al. derive the kinematic equations for the dual quaternion representation of rigid body motions for a strapdown inertial navigation system with highly accurate measurements [7]. In their work, the focus is on accurate integration of inertial measurements and not on sensor fusion. Therefore, the propagation of the uncertainty associated with pose estimates is not described. They give a differential equation to propagate a screw vector during minor time intervals, which is used to update the dual quaternion estimate of the sensor's pose regularly.

In textbooks on GNSS-INS integration, the error model for the IMU's pose often comprises an additive position error and a multiplicative attitude error, thereby treating errors in position and attitude separately and not as elements of the Lie algebra se(3), c.f. [8]. Jones et al. use this approach to describe pose error in their inertial aided visual SLAM system, where an EKF is used to estimate the state [9]. They derive the equations for the time prediction step starting from the body velocity differential equation (8) and end up with a motion model where the velocity is given in the reference frame as it is also described in [8]. In order to improve the observability properties of the state to be estimated, they include the gravity vector in the filter's state and estimate the pose of the sensor system w.r.t. the first sensor pose.

A similar parameterization is chosen by Lupton for a graph-based optimization approach to the integration of inertial and visual measurements [10]. Here, the errors in attitude and position are also treated like independent entries in the state vector and the system's pose is estimated relative to the very first pose with the direction of the gravity vector included in the state.

Recently, another visual-inertial system based on a factor graph formulation of the SLAM problem was presented by Indelman et al. [11]. The authors claim to use a Lie algebra formulation when estimating the sensor's trajectory, but the state prediction equation that relates the IMU's measurements to consecutive sensor poses is not given explicitly.

While applications of the Lie algebra seemingly are rare when it comes to the integration of inertial and visual measurements, a number of authors exploit the Lie algebra formulation in connection with pose graph optimization or bundle adjustment methods that do not require a sophisticated motion model.

For instance, Agrawal exploits the Baker-Campbell-Hausdorff equation that operates on elements of the Lie algebra to formulate a measurement equation that relates adjacent poses to pose-pose constraints, which were derived by visual odometry with a stereo camera system [12]. This approach enables to express poses as well as pose-pose constraints as elements of the Lie algebra se(3), thereby allowing for an elegant formulation of the optimization procedure.

Strasdat et al. present a visual SLAM approach that is based on an efficient bundle adjustment implementation and a graphical interpretation of the dependencies between state variables [13]. By default, camera pose errors and pose-pose

constraints are represented by elements of se(3). However, when closing large loops with a monocular camera the camera poses are are represented by similarity transformations, where one parameter contains a local scale factor and the errors belong to the corresponding Lie group sim(3).

A bundle adjustment framework for SLAM that allows to compare different parameterizations to represent poses is presented in [14]. The authors report improved convergence properties of the iterative minimization procedure when camera pose errors are parameterized as Lie algebra elments as opposed to a parameterization by independent rotation and translation vectors.

IV. EKF-SLAM Formulation

A. Preliminaries

The following sections present the EKF-SLAM formulation that was used for the comparison of different pose error parameterizations presented in sec. V. The overall objective is to estimate both the pose of the IMU's *body coordinate frame* $\{b\}$ and the locations of observed landmarks w.r.t. the *navigation frame* $\{n\}$. The navigation frame is a fix coordinate system with its x- and y-axis aligned with the north and east directions and the z-axis aligned with gravity. However, in order to reduce the effects of linearization errors, an intermediate *strapdown coordinate system* $\{s\}$ is employed, following [10] and [9]. The EKF is then applied to estimate the IMU's pose and the position of observed landmarks in the coordinates of the strapdown coordinate system. Additionally, the pose of the strapdown frame in relation to the navigation frame needs to be estimated. Thus, the EKF state vector is of the form

$$\mathbf{s}_t \quad = \quad \left[\mathbf{s}'^{\mathrm{T}}\ \mathbf{m}^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{9}$$

Where $\mathbf{s}'$ comprises the parameters describing the IMU's pose and motion. Its specific composition depends on the chosen pose error parameterization. Therefore, a more detailed description is deferred until secs. IV-B3 to IV-B5. The Cartesian coordinates of all landmarks that are included in the filter's state are combined in the map vector $\mathbf{m}$:

$$\mathbf{m} \quad = \quad \left[\mathbf{Y}_1^{\mathrm{T}}\ \ldots\ \mathbf{Y}_N^{\mathrm{T}}\right]^{\mathrm{T}} \tag{10}$$

Here, N is the number of landmarks. Subsequently, estimated values are denoted by a hat $\hat{(\cdot)}$ and a tilde $\tilde{(\cdot)}$ indicates the error, i.e., the deviation between a true value $(\cdot)$ and its estimate: $\tilde{(\cdot)} = (\cdot) - \hat{(\cdot)}$.

Because the EKF relies on a truncation of the Taylor series expansion of both the time update step and the measurement equation, it can be regarded as an estimator for the state error, which lies in the tangent plane of the manifold that contains all possible state values. Therefore, the dimension of the error state $\tilde{\mathbf{s}}$ is not necessarily identical to the dimension of the state vector $\mathbf{s}$. In particular, any parameterization for the pose of a rigid body can be chosen, like a homogeneous matrix (1) or a combination of a quaternion and a position vector. Independent of this, the error may be represented by a screw motion or by a rotation vector and a translation vector. Only the covariance matrix, the Jacobians for the measurement as well as the prediction step, and the state update equations after measurements will change depending on the chosen error model.

B. Prediction step

1) Inertial measurements: During the prediction step, the IMU's pose and the associated uncertainty are propagated in time. The IMU measures its acceleration and angular velocity relative to an inertial reference frame:

$$\begin{aligned} {}^{b}_{m}\mathbf{a} &= {}^{b}\mathbf{a} + \mathbf{b}_a + \mathbf{n}_a && (11)\\ {}^{b}_{m}\boldsymbol{\omega} &= {}^{b}\boldsymbol{\omega} + \mathbf{b}_g + \mathbf{n}_g && (12) \end{aligned}$$

Where ${}^{b}_{m}\mathbf{a}$ and ${}^{b}_{m}\boldsymbol{\omega}$ are acceleration and angular rate measurements, respectively. These measurements are disturbed by additive, white Gaussian noise terms $\mathbf{n}_a$ and $\mathbf{n}_g$ as well as additive biases $\mathbf{b}_a$ and $\mathbf{b}_g$. The biases change according to a random walk process driven by white Gaussian noise terms $\mathbf{n}_{b_a}$ and $\mathbf{n}_{b_g}$.

2) Covariance propagation: The error state is propagated according to a first order differential equation that depends on the chosen error model as well as the physical model:

$$\dot{\tilde{\mathbf{s}}}' = F \cdot \tilde{\mathbf{s}}' + G \cdot \mathbf{n} \tag{13}$$

Here, vector $\mathbf{n}$ summarizes the noise terms. The entries of the matrices F and G are determined by the coefficients of the time derivatives given in secs. IV-B3 to IV-B5. With the time derivatives of the error state, the covariance is propagated as follows for each inertial measurement:

$$\begin{aligned} \Phi &= \exp(F \cdot \tau) \approx I_{15\times 15} + F \cdot \tau && (14)\\ P'_{t+\tau} &= \Phi \cdot P_{\tilde{\mathbf{s}}',\tilde{\mathbf{s}}'} \cdot \Phi_t^{\mathrm{T}} + \Phi \cdot G \cdot Q \cdot G^{\mathrm{T}} \cdot \Phi^{\mathrm{T}} \tau && (15)\\ P_{t+\tau} &= \begin{bmatrix} P'_{t+\tau} & \Phi \cdot P_{\tilde{\mathbf{s}}',\tilde{\mathbf{m}}} \\ P_{\tilde{\mathbf{m}},\tilde{\mathbf{s}}'} \cdot \Phi^{\mathrm{T}} & P_{\tilde{\mathbf{m}},\tilde{\mathbf{m}}} \end{bmatrix} && (16) \end{aligned}$$

In the expression above, P and P' are the covariance matrices pertaining to $\mathbf{s}$ and $\mathbf{s}'$, respectively. Furthermore, Q is the power spectral density matrix which characterizes the noise vector $\mathbf{n}$.

3) Standard error state: In this work, the term *standard error state* refers to the following description of the relationship between the true IMU pose T_b^s and its estimate $\hat{T}_b^s$:

$$T_b^s \quad = \quad \begin{bmatrix} C(\boldsymbol{\Psi}_b^s) \cdot \hat{C}_b^s & {}^{s}\hat{\mathbf{p}}_b + {}^{s}\tilde{\mathbf{p}}_b \\ 0 & 1 \end{bmatrix} \tag{17}$$

Where $C(\boldsymbol{\Psi}) = \exp(\boldsymbol{\Psi})$ is the rotation matrix corresponding to the Rodrigues vector $\boldsymbol{\Psi} \in so(3)$. The definition (17) is equivalent to the definition given in [8]. In this case, the error state is given by:

$$\tilde{\mathbf{s}}' \quad = \quad \left[{}^{s}\tilde{\mathbf{p}}_b^{\mathrm{T}}\ {}^{s}\tilde{\mathbf{v}}_b^{\mathrm{T}}\ \tilde{\mathbf{b}}_a^{\mathrm{T}}\ \boldsymbol{\Psi}_b^{s\mathrm{T}}\ \tilde{\mathbf{b}}_g^{\mathrm{T}}\ {}^{n}\tilde{\mathbf{p}}_s^{\mathrm{T}}\ \boldsymbol{\Psi}_s^{n\mathrm{T}}\right]^{\mathrm{T}} \tag{18}$$

In order to propagate the estimate of the IMU's pose in time, the effect of gravity on the inertial measurements needs to be compensated. This leads to the following equations to update the velocity and pose estimates:

$$ {}^s\hat{\mathbf{a}} = \hat{C}_b^s \cdot ({}_m^b\mathbf{a} - \hat{\mathbf{b}}_a) + \hat{C}_n^s \cdot {}^n\mathbf{g} \tag{19} $$

$$ {}^s\hat{\mathbf{p}}_{b,t+\tau} = {}^s\hat{\mathbf{p}}_b + {}^s\hat{\mathbf{v}}_b \cdot \tau + \frac{1}{2} {}^s\hat{\mathbf{a}} \cdot \tau^2 \tag{20} $$

$$ {}^s\hat{\mathbf{v}}_{b,t+\tau} = {}^s\hat{\mathbf{v}}_b + {}^s\hat{\mathbf{a}} \cdot \tau \tag{21} $$

$$ \hat{C}_{b,t+\tau}^s = \hat{C}_b^s \cdot C({}^b\hat{\boldsymbol{\omega}} \cdot \tau) \tag{22} $$

Where τ is the time interval between consecutive inertial measurements. Thus, the time derivatives, which are needed to propagate the covariance according to (13), can now be stated as:

$$ \dot{{}^s\tilde{\mathbf{p}}}_b = {}^s\tilde{\mathbf{v}}_b \tag{23} $$

$$ \dot{{}^s\tilde{\mathbf{v}}}_b = \left\lfloor -\hat{C}_b^s \cdot ({}_m^b\mathbf{a}_b - \hat{\mathbf{b}}_a) \right\rfloor_\times \cdot \boldsymbol{\Psi}_b^s + \hat{C}_b^s \cdot \mathbf{n}_a - \ldots \\ \hat{C}_b^s \cdot \tilde{\mathbf{b}}_a + \hat{C}_s^{n\mathrm{T}} \cdot \lfloor {}^n\mathbf{g} \rfloor_\times \cdot \boldsymbol{\Psi}_s^n \tag{24} $$

$$ \dot{\tilde{\mathbf{b}}}_a = \mathbf{n}_{b_a} \tag{25} $$

$$ \dot{\boldsymbol{\Psi}}_{\mathbf{b}}^{\mathbf{s}} = -\hat{C}_b^s \cdot (\tilde{\mathbf{b}}_g + \mathbf{n}_g) \tag{26} $$

$$ \dot{\tilde{\mathbf{b}}}_g = \mathbf{n}_{b_g} \tag{27} $$

$$ \dot{{}^n\tilde{\mathbf{p}}}_s = 0_{3\times3} \tag{28} $$

$$ \dot{\boldsymbol{\Psi}}_s^n = 0_{3\times3} \tag{29} $$

Henceforth, the error model given by (17) is always used for the transformation between the $\{s\}$ and $\{n\}$ frames. Since it does not change in time, its time derivatives (28) and (29) vanish and are thus omitted in the ensuing sections. Moreover, the time derivatives for the bias terms (25) and (27) as well as the attitude propagation equation (22) will not be restated.

4) Left-invariant error twist: When using a multiplicative error model based on the twist representation, the corrections can be applied either by multiplying from the left or from the right. The latter case is subsequently called *left-invariant* error twist, because the error is invariant w.r.t. multiplications by constant matrices from the left. Consequently, applying the correcting transformations from the left is henceforth referred to as *right-invariant*.

For the left-invariant case the corrections are applied as follows:

$$ \begin{aligned} T_b^s &= \hat{T}_b^s \cdot \exp(S_{\boldsymbol{\Psi},\boldsymbol{\xi}}) \\ &= \begin{bmatrix} \hat{C}_b^s \cdot C(\boldsymbol{\Psi}) & \hat{C}_b^s \cdot \mathbf{u} + {}^s\hat{\mathbf{p}}_b \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{30} $$

With $C(\boldsymbol{\Psi})$ and $\mathbf{u}$ defined by:

$$ \exp(S_{\boldsymbol{\Psi},\boldsymbol{\xi}}) = \begin{bmatrix} C(\boldsymbol{\Psi}) & \mathbf{u} \\ 0 & 1 \end{bmatrix} \tag{31} $$

Hence, the error model is different from (17). In particular, the position correction has to be transformed to the reference system. Note, that the error twist in (30) is similar to a body velocity as defined in (8). It is assumed that the IMU moves according to

$$ \dot{T}_b^s = T_b^s \cdot S_{\boldsymbol{\omega},\mathbf{v}}. \tag{32} $$

Where $S_{\boldsymbol{\omega},\mathbf{v}}$ is the body velocity. The error state vector for this parameterization is:

$$ \tilde{\mathbf{s}}' = \left[\boldsymbol{\xi}^{\mathrm{T}}\ {}^b\tilde{\mathbf{v}}^{\mathrm{T}}\ \tilde{\mathbf{b}}_a^{\mathrm{T}}\ \boldsymbol{\Psi}_b^{s\mathrm{T}}\ \tilde{\mathbf{b}}_g^{\mathrm{T}}\ {}^n\tilde{\mathbf{p}}_s^{\mathrm{T}}\ \boldsymbol{\Psi}_s^{n\mathrm{T}} \right]^{\mathrm{T}} \tag{33} $$

Here, the position error and the translational velocity relative to the reference frame are replaced by the corresponding part of the error twist and the body velocity. Subsequently, the shorthand notation $E(\boldsymbol{\Psi},\boldsymbol{\xi}) = \exp(S_{\boldsymbol{\Psi},\boldsymbol{\xi}})$ is used for the multiplicative error. Also, the error twist and the body velocity twist are written as vectors $\mathbf{t}_\epsilon$ and $\mathbf{t}_v$, respectively. Solving (30) for the error and taking the time derivative gives:

$$ \dot{E}(\boldsymbol{\Psi},\boldsymbol{\xi}) = \left((\hat{T}_b^s)^{-1} \cdot \dot{T}_b^s \right) \tag{34} $$

$$ = -S_{\hat{\boldsymbol{\omega}},\hat{\mathbf{v}}} \cdot (\hat{T}_b^s)^{-1} \cdot T_b^s + (\hat{T}_b^s)^{-1} \cdot \dot{T}_b^s \cdot S_{\boldsymbol{\omega},\mathbf{v}} \tag{35} $$

$$ = -S_{\hat{\boldsymbol{\omega}},\hat{\mathbf{v}}} \cdot E(\boldsymbol{\Psi},\boldsymbol{\xi}) + E(\boldsymbol{\Psi},\boldsymbol{\xi}) \cdot S_{\boldsymbol{\omega},\mathbf{v}} \tag{36} $$

In [15], Bullo and Murray derive a version of this differential equation that only depends on elements in the Lie algebra. If second order terms, i.e., products of error terms, are neglected, this can be stated as:

$$ \dot{\mathbf{t}}_\epsilon = -\mathrm{ad}_{\hat{t}_v}(\mathbf{t}_\epsilon) + \mathbf{t}_v - \hat{\mathbf{t}}_v \tag{37} $$

Or, equivalently:

$$ \begin{bmatrix} \dot{\boldsymbol{\Psi}} \\ \dot{\boldsymbol{\xi}} \end{bmatrix} = \begin{bmatrix} -\lfloor {}^b\hat{\boldsymbol{\omega}} \rfloor_\times & 0 \\ -\lfloor {}^b\hat{\mathbf{v}} \rfloor_\times & -\lfloor {}^b\hat{\boldsymbol{\omega}} \rfloor_\times \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\Psi} \\ \boldsymbol{\xi} \end{bmatrix} + \begin{bmatrix} {}^b\tilde{\boldsymbol{\omega}} \\ {}^b\tilde{\mathbf{v}} \end{bmatrix} \tag{38} $$

Eq. (38) describes the change of the error state in time for the twist representation. With this relationship the time derivatives for the whole error state can be written as follows:

$$ \dot{\boldsymbol{\xi}} = -\lfloor {}^b\hat{\mathbf{v}} \rfloor_\times \cdot \boldsymbol{\Psi}_b^s - \lfloor {}^b\hat{\boldsymbol{\omega}} \rfloor_\times \cdot \boldsymbol{\xi} + {}^b\tilde{\mathbf{v}} \tag{39} $$

$$ \dot{{}^b\tilde{\mathbf{v}}} = C(-{}^b\hat{\boldsymbol{\omega}} \cdot \tau) \cdot \left(\mathbf{n}_a - \tilde{\mathbf{b}}_a + \left\lfloor \hat{C}_b^{s\mathrm{T}} \cdot \hat{C}_s^{n\mathrm{T}} \cdot {}^n\mathbf{g} \right\rfloor_\times \cdot \boldsymbol{\Psi}_b^s + \ldots \\ \hat{C}_b^{s\mathrm{T}} \cdot \hat{C}_s^{nT} \cdot \lfloor {}^n\mathbf{g} \rfloor_\times \cdot \boldsymbol{\Psi}_s^n \right) - \lfloor {}^b\hat{\boldsymbol{\omega}} \rfloor_\times {}^b\tilde{\mathbf{v}} \tag{40} $$

$$ \dot{\boldsymbol{\Psi}}_b^s = -\lfloor {}^b\hat{\boldsymbol{\omega}} \rfloor_\times \cdot \boldsymbol{\Psi}_b^s - \tilde{\mathbf{b}}_g - \mathbf{n}_g \tag{41} $$

The corresponding equations to propagate the position and velocity estimates in time are:

$$
\begin{aligned}
{}^{b}\hat{\mathbf{a}} \quad &= {}^{b}_{m}\mathbf{a} - \hat{\mathbf{b}}_a + \hat{C}^{b}_{s} \cdot \hat{C}^{s}_{n} \cdot {}^{n}\mathbf{g} && (42)\\
{}^{s}\hat{\mathbf{p}}_{b,t+\tau} &= {}^{s}\hat{\mathbf{p}}_b + \hat{C}^{s}_{b}\left({}^{b}\hat{\mathbf{v}}_b \cdot \tau + \frac{1}{2}{}^{b}\hat{\mathbf{a}} \cdot \tau^2\right) && (43)\\
{}^{b}\hat{\mathbf{v}}_{t+\tau} &= C(-{}^{b}\hat{\boldsymbol{\omega}} \cdot \tau) \cdot ({}^{b}\hat{\mathbf{v}} + {}^{b}\hat{\mathbf{a}} \cdot \tau) && (44)
\end{aligned}
$$

5) Right-invariant error twist: Applying the corrections to the estimated pose by left multiplication leads to the following update rule:

$$
\begin{aligned}
T^{s}_{b} &= \exp(S_{\boldsymbol{\Psi},\boldsymbol{\xi}}) \cdot \hat{T}^{s}_{b} \\
&= \begin{bmatrix} C(\boldsymbol{\Psi}) \cdot \hat{C}^{s}_{b} & \mathbf{u} + C(\boldsymbol{\Psi}) \cdot {}^{s}\hat{\mathbf{p}}_b \\ 0 & 1 \end{bmatrix} && (45)
\end{aligned}
$$

In this case, attitude corrections also affect the position estimate. In conjunction with this error model the subsequent error state vector is used:

$$
\tilde{\mathbf{s}}' = \left[\boldsymbol{\xi}^{\mathrm{T}}\ {}^{s}\tilde{\mathbf{v}}^{\mathrm{T}}_{b}\ \tilde{\mathbf{b}}^{\mathrm{T}}_{a}\ \boldsymbol{\Psi}^{s\mathrm{T}}_{b}\ \tilde{\mathbf{b}}^{\mathrm{T}}_{g}\ {}^{n}\tilde{\mathbf{p}}^{\mathrm{T}}_{s}\ \boldsymbol{\Psi}^{n\mathrm{T}}_{s}\right]^{\mathrm{T}} \quad (46)
$$

The parameters $\boldsymbol{\Psi}^{s}_{b}$ and $\boldsymbol{\xi}$ now belong to the spatial error twist (45). A derivation as described in the previous section yields the time derivative of the error state if a spatial velocity twist is used in conjunction with the spatial error twist. However, the calculations are more involved and a larger number of second order terms were omitted during the calculations. Finally, the time derivatives can be stated:

$$
\begin{aligned}
\dot{\boldsymbol{\xi}} &= \lfloor {}^{s}\mathbf{v}_b \rfloor_{\times} \cdot \boldsymbol{\Psi}^{s}_{b} - \lfloor {}^{s}\hat{\mathbf{p}}_b \rfloor_{\times} \cdot \hat{C}^{s}_{b} \cdot (\mathbf{b}_g + \mathbf{n}_r) + {}^{s}\tilde{\mathbf{v}}_b && (47)\\
{}^{s}\dot{\tilde{\mathbf{v}}}_b &= \left[-\hat{C}^{s}_{b} \cdot ({}^{b}_{m}\mathbf{a}_b - \hat{\mathbf{b}}_a)\right]_{\times} \cdot \boldsymbol{\Psi}^{s}_{b} + \hat{C}^{s}_{b} \cdot \mathbf{n}_a - \ldots \\
&\quad \hat{C}^{s}_{b} \cdot \tilde{\mathbf{b}}_a + \hat{C}^{n\mathrm{T}}_{s} \cdot \lfloor {}^{n}\mathbf{g} \rfloor_{\times} \cdot \boldsymbol{\Psi}^{n}_{s} && (48)\\
\dot{\boldsymbol{\Psi}}^{s}_{b} &= -\hat{C}^{s}_{b} \cdot (\tilde{\mathbf{b}}_g + \mathbf{n}_g) && (49)
\end{aligned}
$$

Eqs. (19)-(22) are used to propagate the estimated pose in time.

C. Measurement update

Characteristically textured surface elements serve as landmarks whose projections onto the image plane are continuously tracked in the video stream. The image coordinates of these projections are are stacked to a single measurement vector $\mathbf{z}$ that is used to update the filter state by performing an EKF update step to obtain an estimate of the error state $\tilde{\mathbf{s}}$, c.f. [16]:

$$
\begin{aligned}
K &= H^{\mathrm{T}} \cdot P \cdot (H \cdot P \cdot H^{\mathrm{T}} + R)^{-1} && (50)\\
\hat{\tilde{\mathbf{s}}} &= K \cdot (\mathbf{z} - h(\mathbf{s})) && (51)
\end{aligned}
$$

Here, R is the covariance matrix of the stacked measurement vector and H is the Jacobian of the measurement equation, which consists of a coordinate transformation followed by a central projection. The linearized measurement equation for a landmark $\mathbf{Y}_i$ can be stated as

$$
\mathbf{z}_i = h(\hat{\mathbf{s}}', \hat{\mathbf{Y}}_i) + H_{\mathbf{s}'} \cdot \tilde{\mathbf{s}}' + H_{\mathbf{y}} \cdot \tilde{\mathbf{y}}_y + \mathbf{v}\ , \quad (52)
$$

thereby making its dependence on the error state explicit. In (52) $\mathbf{v}$ is the zero mean measurement noise. The Jacobians $H_{\mathbf{s}'}$ and $H_{\mathbf{y}}$ depend on the chosen error model. They are calculated numerically by applying small perturbations to the estimated states according to the multiplicative error models described in sec. IV-B3 to IV-B5. Finally, the estimated error state is employed to update the state variables, again by applying the appropriate update rule for the pose estimates. In addition, the covariance matrix is updated according to:

$$
P^{+} = P^{-} - K \cdot (H \cdot P^{-} \cdot H^{\mathrm{T}} + R)^{-1} \cdot K^{\mathrm{T}} \quad (53)
$$

These measurement update equations in connection with the equations for the time update given in the previous section define the EKF-SLAM estimator for different pose parameterizations.

When new landmarks are introduced in the filter's state, an inverse form of the measurement model is employed in order to obtain an initial estimate of their position and the cross covariance terms. New landmarks are initialized with a fix depth of eight meters and a large uncertainty in the direction of the projection ray.

Both the position vector and the yaw angle pertaining to the transformation T^{n}_{s} are not observable, c.f. [9]. For the simulation experiments the corresponding entries in the state vector were therefore fixed by setting the respective entries in the covariance matrix and calculated Jacobians to zero.

On the contrary, when processing real data sequences, we make use of the composition step to reduce the effects of linearization related errors as originally proposed in [17]. The composition step is essentially a coordinate transformation that is applied to reset the reference frame. In this case the complete transformation T^{n}_{s} is needed to perform the composition step.

V. Experimental Results

A. Simulation runs

A number of simulation runs were conducted in order to compare the different approaches to pose error parameterization. For this purpose, acceleration measurements were generated by sampling the second derivative of a C^2-spline that is used to specify the trajectory of the sensor system. Similarly, angular rate measurements were obtained from the incremental rotations between successive reference frames. White, Gaussian noise with zero mean and additive biases were added to the generated inertial measurements. The parameters of the artificial noise were chosen to imitate the noise characteristics measured by the sensor system used for the experiments described in V-B. Moreover, landmark observations were simulated by projecting known reference points onto the image plane and adding Gaussian noise with zero mean and a standard deviation of one pixel.

The simulations provide accurate ground truth which allows to assess the consistency of the employed filtering technique by computing the NEES measure [18]. For a consistent estimator, the NEES should be roughly below a threshold of 14 most of the time.

Fig. 2 presents the results for a simulated walk through a long rectangular-shaped hallway with sharp right turns at the end of each section. In this scene, landmarks go out of view while new landmarks have to be added to the filter's state regularly.

According to the plots for pose error and NEES in Fig. 2b and 2c, the right invariant error parameterization performs very similar to the standard error parameterization concerning attitude error as well as attitude NEES while the plots for the left-invariant error deviate from this pattern. However, the situation changes for the position estimates. Here, the left-invariant and the standard error parameterization are almost indistinguishable while the plots for the right-invariant parameterization are slightly different. At the end of the hallway trajectory, the standard error parameterization slightly outperforms the two alternative approaches.

This behavior may be explained by the structure of the update equations (17), (30), and (45): The attitude correction equations are identical for the right-invariant and the standard parameterization, but the position update for the right-invariant parameterization also depends on the estimated attitude error.

Because the plots are partly indistinguishable, tables I and II additionally summarize the simulation results in terms of average pose error and average NEES.

Overall, the results for the simulated turntable sequence shown in Fig. 3 confirm these findings. Yet, the standard error state and the left-invariant error parameterization are even closer regarding position error and position consistency. This indicates, that the slight difference between them observed at the end of the hallway sequence is probably mainly due to linearization errors which are introduced when new landmarks are included. These play a less important role in the turntable sequence, because new landmarks are only inserted once in the beginning.

The NEES plots show that the filter is inconsistent for all parameterizations under investigation. The inconsistency mainly stems from the fact that landmarks are initialized with a fix depth and a large uncertainty because the Gaussian distribution assumption is severely violated thereby.

B. Indoor experiment

The EKF-SLAM approach was also applied to an indoor dataset that was recorded in an office building using a sensor system fixed to the torso of a pedestrian. The employed sensor system is composed of MEMS accelerometers with 5-10 mg RMS noise characteristics and gyroscopes that are subject to a 0.0056 $^\circ$ / (sec $\cdot\sqrt{\text{sec}}$) angular velocity random walk according to the manufacturers. In addition, a camera records video images with a resolution of 1398x1080 pixels at 28 Hz. These images are scaled down to half size and the Harris corner detector with subsequent subpixel estimation is employed to detect salient image points which are tracked with a pyramidal implementation of the Lucas-Kanade algorithm [19], [20].

The algorithm was run on this indoor dataset once for each of the three presented error parameterizations performing a composition step regularly as described in sec. IV-C. The calculated trajectories are shown in Fig. 4 printed on top of the building's floor plan. Since accurate ground truth is not available for this dataset, a comparison can only be made by

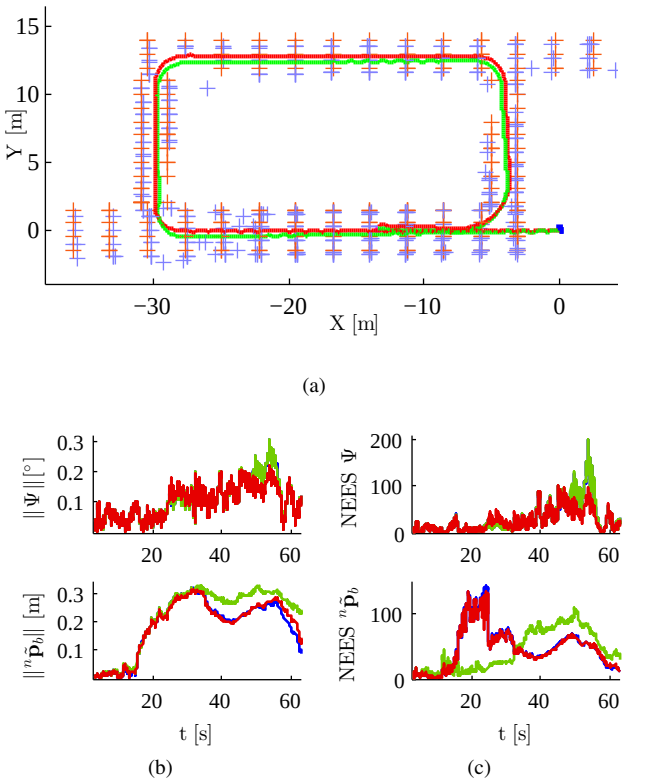

Fig. 2. Simulation results for hallway sequence. (a) Top view of a simulation run with the standard error state parameterization. Green: Estimated trajectory Light blue: Estimated landmark positions Red: Reference trajectory Orange: Reference landmark positions. (b) Attitude (top) and position (bottom) error. (c) Attitude (top) and position (bottom) NEES. Color codings for (b) and (c): Green: Right-invariant error Red: Left-invariant error Blue: Standard error state.

TABLE I
SIMULATION RESULTS: HALLWAY SEQUENCE

	Avg. NEES position	Avg. NEES attitude	Avg. error position [m]	Avg. error attitude [°]
Left-invariant	45.5793	24.2785	0.1895	0.1015
Right-invariant	42.644443	29.209042	0.2192	0.1081
Standard Error-state	46.7471	28.9725	0.1876	0.1077

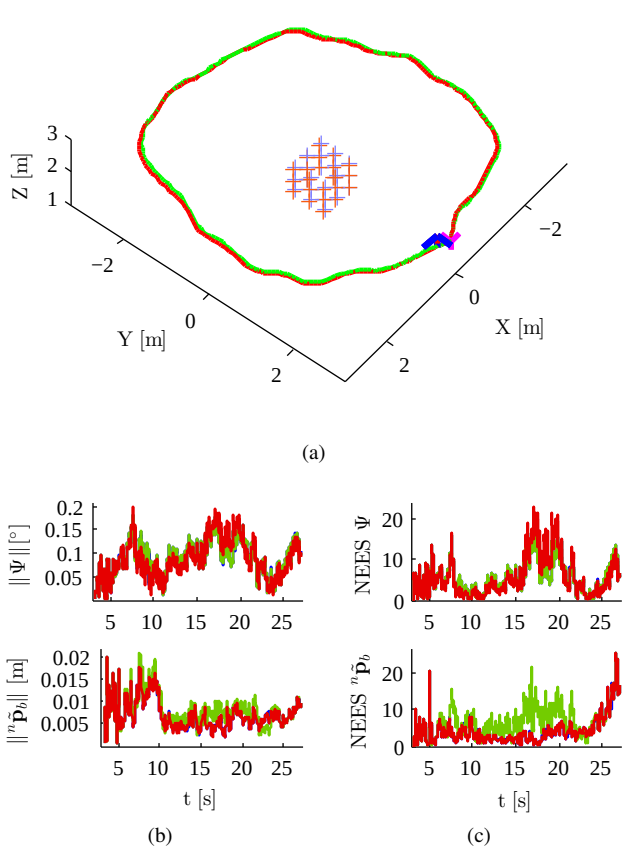

Fig. 3. Simulation results for cube sequence. (a) Slanted view of a simulation run with the standard error state parameterization. (b) Attitude (top) and position error (bottom). (c) Attitude (top) and position (bottom) NEES. See Fig. 2 for details.

TABLE II
SIMULATION RESULTS: CUBE SEQUENCE

	Avg. NEES position	Avg. NEES attitude	Avg. error position [m]	Avg. error attitude [°]
Left-invariant	4.0015	5.5863	0.0070	0.0881
Right-invariant	6.6691	5.3010	0.0071	0.0896
Standard Error-state	4.0421	5.3024	0.0070	0.0896

comparing the estimated trajectories with the floor plan. An integrated compass was used to determine the initial attitude of the sensor system w.r.t. the map and the starting position was chosen such that the remaining trajectory fits best to the floor plan.

It can be observed that the trajectories estimated with the different error parameterizations increasingly differ from each other the farther the system gets from the starting point. However, the results obtained with all parameterizations are

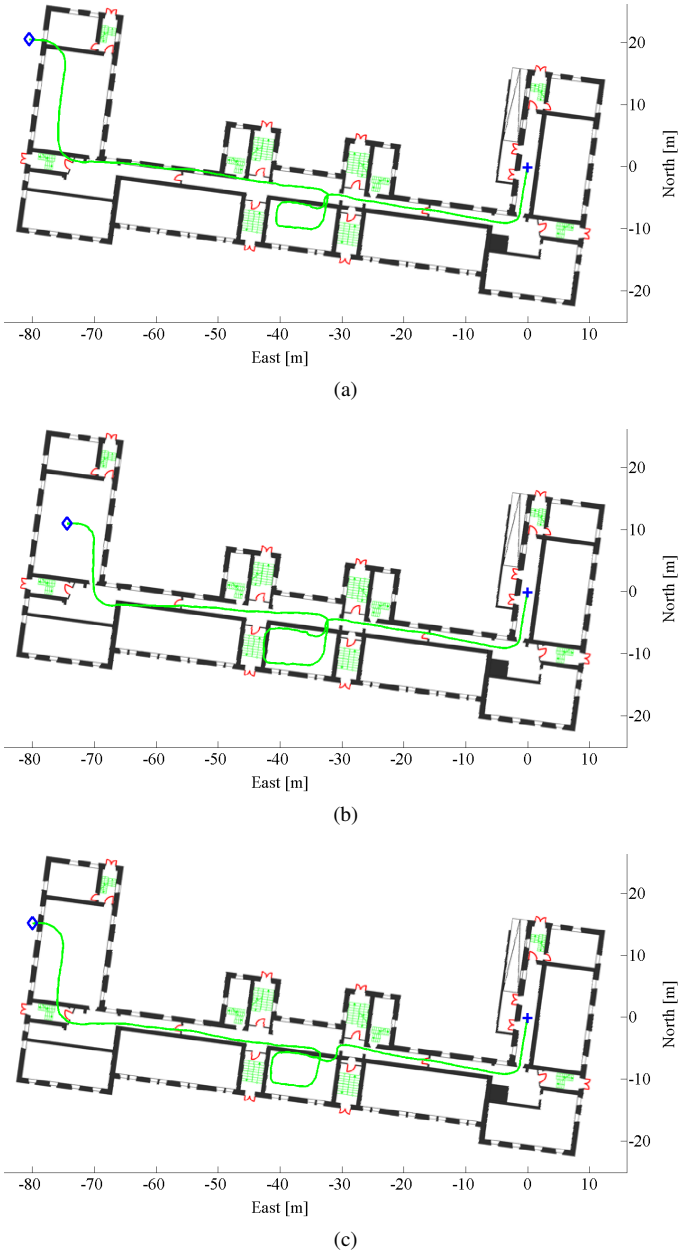

Fig. 4. Results on indoor dataset. Top: Left-invariant error twist, Center: Right-invariant error twist, Bottom: Standard error state. A blue cross marks the starting position. A blue diamond marks the trajectory's end.

very similar in quality. In addition, it is not clear whether the observed differences primarily stem from the chosen pose error model or from linearization related inaccuracies pertaining to the initialization process for new landmarks, which is also different for each error model. Thus, it is not possible to determine which of the compared parameterizations is best based on this experiment.

VI. CONCLUSION

This work compares different pose error parameterizations for inertial aided visual SLAM. It is noted, that the widespread error state formulation reflects the structure of the double-

geodesics on the cross-product space $\mathrm{SO}(3) \times \mathbb{R}^3$ while the twist representation more closely resembles the mathematical structure of the associated group $\mathrm{SE}(3)$. The error state parameterizations and the corresponding time and measurement update equations for the applied EKF are described.

A comparison of the different parameterizations is presented based on simulated trajectories and a real indoor dataset. For the simulated trajectories, the differences between the approaches regarding pose error and consistency are marginal. However, the standard error state approach seems to be slightly beneficial. Similarly, the results on the real indoor dataset do not clearly favor one parameterization.

Its incapability to relinearize about past states is a well known disadvantage of the EKF employed for state estimation in this work. It would therefore be of interest to compare the presented motion and measurement models in a batch estimation framework with relinearization in future work in order to investigate to what extend the different parameterizations can take advantage of relinearization.

ACKNOWLEDGMENT

The authors would like to thank Mr. Christian Ascher from the Institute of Systems Optimization (ITE) at the Karlsruhe Institute of Technology (KIT) for software and hardware collaboration.

REFERENCES

[1] R. McCroskey, P. Samanant, W. Hawkinson, S. Huseth, and R. Hartman, "Glanser - an emergency responder locator system for indoor and gps-denied applications," in *23rd International Technical Meeting of the Satellite Division of The Institute of Navigation, Portland, OR, September 21-24, 2010*, 2010.

[2] J. Selig, *Geometric Fundamentals of Robotics (Second Edition)*. Springer, 2005.

[3] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[4] F. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," *Journal of Mechanical Design*, vol. 117, pp. 48–54, 1995.

[5] L. Dorst, *The Representation of Rigid Body Motions in the Conformal Model of Geometric Algebra*. Springer, 2008, ch. 21, pp. 507–529.

[6] R. Mahony and J. H. Manton, "The geometry of the newton method on non-compact lie groups," *Journal of Global Optimization*, vol. 23, pp. 309–327, 2002.

[7] Y. Wu, X. Hu, D. Hu, T. Li, and J. Lian, "Strapdown inertial navigation system algorithms based on dual quaternions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 110–132, 2005.

[8] J. Farrell and M. Barth, *The Global Positioning System & Inertial Navigation*. McGraw-Hill, 1999.

[9] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011. [Online]. Available: http://ijr.sagepub.com/content/30/4/407.abstract

[10] T. Lupton, "Inertial slam with delayed initialisation," Ph.D. dissertation, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, 2010.

[11] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Factor graph based incremental smoothing in inertial navigation systems," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, 2012.

[12] M. Agrawal, "A lie algebraic approach for consistent pose registration for motion estimation," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[13] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual slam," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011.

[14] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 3607 –3613.

[15] F. Bullo and R. M. Murray, "Proportional derivative (pd) control on the euclidean group," California Institute of Technology, Division of Engineering and Applied Science, Tech. Rep., 1995.

[16] A. Gelb, *Applied Optimal Estimation*. The MIT Press, ISBN: 0262570483, 1995.

[17] J. A. Castellanos, R. Martinez-Cantin, J. D. Tardós, and J. Neira, "Robocentric map joining: Improving the consistency of ekf-slam," *Robotics and Autonomous Systems*, vol. 55, pp. 21–29, 2007.

[18] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.

[19] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, 1988.

[20] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm," Intel Corporation Microprocessor Research Labs, Tech. Rep., 2002.