Robotica (2015) volume 33, pp. 1250–1280. © Cambridge University Press 2014
doi:10.1017/S0263574714000678

Quaternion-based EKF-SLAM from relative pose measurements: observability analysis and applications

Luca Carlone†,* Vito Macchia‡, Federico Tibaldi‡ and Basilio Bona§

†*College of Computing, Georgia Institute of Technology, Atlanta, USA*
‡*Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy*
§*Istituto Superiore Mario Boella, Torino, Italy*

(Accepted February 17, 2014. First published online: April 1, 2014)

SUMMARY
In this work, we investigate a quaternion-based formulation of 3D Simultaneous Localization and Mapping with Extended Kalman Filter (EKF-SLAM) using relative pose measurements. We introduce a discrete-time derivation that avoids the *normalization problem* that often arises when using unit quaternions in Kalman filter and we study its observability properties. The consistency of the estimation errors with the corresponding covariance matrices is also evaluated. The approach is further tested on real data from the *Rawseeds dataset* and it is applied within a delayed-state EKF architecture for estimating a dense 3D map of an unknown environment. The contribution is motivated by the possibility of abstracting multi-sensorial information in terms of relative pose measurements and for its straightforward extensions to the multi robot case.

KEYWORDS: 3D EKF-SLAM; Unit quaternions; Observability; Relative pose measurements; Delayed-state filters.

1. Introduction

Recent developments in robotic navigation are enabling autonomous operation of mobile platforms acting in unknown, unstructured environments.[56] In several application scenarios, ranging from exploration to search and rescue, the construction of a world model and the concurrent estimation of robot location are crucial for mission accomplishment, for enhancing motion planning effectiveness and for attaining a desired level of situational awareness. This estimation problem is usually referred to as *Simultaneous Localization and Mapping* (SLAM). SLAM has been extensively studied in planar scenarios, since the publication of the pioneering papers.[25,57] Extended Kalman Filter (EKF) has been demonstrated to provide an effective solution to the estimation problem: EKF is used as an observer of a nonlinear dynamical system modeling robot motion and sensor perception. A general overview on both theoretical and implementation aspects of SLAM can be found in the survey papers,[23,24] and the references therein.

In the last decade, the diffusion of autonomous robots acting on uneven terrains and the widespread applications of Unmanned Autonomous Vehicles (UAV) for surveillance and monitoring have stressed the importance of extending the SLAM landscape to three-dimensional scenarios. The complexity, in this setup, stems from the larger dimension of the state vectors and from the need of introducing more complex representations of robot attitude. Moreover, the huge amount of data, gathered through 3D sensors, needs to be properly synthesized (*information synthesis*) in order to avoid cumbersome costs in terms of computation and memory requirements.

* Corresponding author. E-mail: luca.carlone@gatech.edu

In classic EKF-SLAM implementations the state vector comprises robot pose at current time and the *position* of *landmarks* in the environment. In this case, the *information synthesis* phase consists in extracting the position of distinguishable landmarks (e.g., points, corners, positions of 3D bodies), that has to be included in the state vector. Recent examples of this strategy are the articles.[2,9,62] The complexity of the filter is quadratic in the number of *landmarks*, hence a naive implementation prevents large scale mapping due to the overwhelming computational cost.[23] A considerable research effort has been devoted to the reduction of the complexity of the approach, see.[26,51] Moreover, several authors proposed the use of the *information form* of the filter, to improve the computational effort of the update phase.[59] It is worth noticing that including the position of each landmark in the state vectors requires to select few significative elements of the environment to be modeled, whereas it may be desirable to have a dense representation of the environment, which can be a more reliable basis for path planning and model-based navigation. Finally, we remark that the effects of partial observability in standard EKF-SLAM implementation are not fully understood, this being witnessed by recent results.[34]

Another interpretation of EKF-SLAM leads to include in the state vector (instead of the point features) the vantage poses of the robot (*observation poses*), from which such features are detected. This allows to make the state dimension independent on the number of features observed from a single observation pose. The corresponding filter is usually referred to as *delayed-state* EKF-SLAM,[19] *view-based* EKF[28] or *pose-based* SLAM.[36]

The contribution of this paper is threefold. First of all, we report a quaternion-based formulation of EKF-SLAM; both prediction and update phase of our EKF are based on relative pose measurements, and the complete derivation of the equations is reported in the paper. The derivation is not based on heuristic solutions for preserving the unit norm of quaternions of rotation (e.g., *normalization*[52]), nor based on the *projection filter*,[47] which introduces spurious measurements to force the unitary norm constraint. We devise filter equations in order to intrinsically preserve the unit norm and to keep the orientation estimates on the manifold SO(3). The effectiveness of the quaternions of rotation lies in their robustness (i.e., *singularity-free*), since no three-parameters representation for rotation (e.g., roll-pitch-yaw or Euler angles[19]) can be both global and non-singular.[41] As a second contribution we present an observability study of the proposed SLAM model. We show that the system is not *locally observable*, although it is *completely observable* over a given interval of time, as long as this interval includes an observation of the *anchor*, i.e., the landmark that was set at the origin of the reference frame for mapping. Consistency of the estimation errors with the derived covariance matrices is also investigated through a numerical analysis and confirms the observability findings. The third contribution is an extensive testing of the proposed approach on both real and simulated data. We discuss the use of the filter in real applications and we show that the approach is accurate in practice; moreover, the proposed filter enables the construction of dense 3D maps of the scenario, as well as sparse 3D landmark-based map models.

The paper extends the results presented in ref. [13] and is structured as follows. Section 2 reviews the literature on 3D SLAM and observability analysis in SLAM. Section 3 reports an informal statement of the problem, while Section 4 presents the equations of the proposed EKF-based approach. We present our observability analysis in Section 5. Applications on real and simulated data are described in Section 6; conclusions are drawn in Section 7. For reader's convenience, Appendix A contains some preliminary concepts on quaternions and rotations.

Notation. Matrices are represented as bold upper case, e.g., $\boldsymbol{M}$, column vectors as bold lower case, e.g., $\boldsymbol{x}$. The symbol $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix, $\mathbf{0}_{n,m}$ denotes a matrix with all zero entries with n rows and m columns. The cardinality of a generic set S is written as $|S|$. The symbols $\oplus$ are $\ominus$ denote the sum and the subtraction in homogeneous coordinates, respectively. The estimate of a generic variable $\boldsymbol{x}$ is represented by a hat variable, $\hat{\boldsymbol{x}}$, whereas a measured quantity is represented by a bar variable, $\bar{\boldsymbol{x}}$; estimation or measurement errors are denoted by tilde variables, $\tilde{\boldsymbol{x}}$. In particular, for Cartesian variables, the estimation error is defined as $\tilde{\boldsymbol{t}} = \boldsymbol{t} - \hat{\boldsymbol{t}}$, whereas the error quaternion is defined as $\tilde{\boldsymbol{q}} = \hat{\boldsymbol{q}}^{-1}\boldsymbol{q}$. Measurement errors are defined accordingly.

2. Related Work

In the introduction we gave a general context for the proposed approach, while in this section we discuss how the contribution extends the literature on 3D SLAM. We classify related work in three

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

main categories, according to the main focus of the articles: *observability*, *3D SLAM*, and *delayed-state filtering*.

The consistency problem in SLAM has been mainly studied in planar setups. Julier and Uhlmann first observed that using standard EKF-SLAM (i.e., state vector comprising robot pose and landmark positions), leads to an artificial reduction of the absolute orientation variance when observing a new landmark.[37] Subsequent analysis by Castellanos *et al.*[15,16] and Huang *et al.*[35] confirmed such observation devising some causes of inconsistency in the observability properties of the SLAM system. Bailey[4] provided numerical evidence that the main cause of inconsistency in standard EKF-SLAM is the uncertainty in robot orientation. The first comprehensive theoretical treatment of the topic traces back to,[34] in which the authors formally showed that the linearization in EKF-SLAM has an observable subspace of higher dimension than the one of the original nonlinear system (the nonlinear system has an unobservable subspace of dimension 3 in a planar case, due to the lack of absolute information). Finally, in ref. [34], the authors presented a solution to address the mismatch between the observability properties of the original system and the ones of the piecewise linear system used in EKF-SLAM.

Observability analysis in 3D problems has been mainly studied in the aerospace literature. In refs. [30, 31] the authors presented an observability study for vehicle navigation based in inertial navigation systems. In ref. [39] it is studied observability in 3D SLAM, using Euler angles, concluding that the system is only partially observable when including only landmark positions in the state vector. The work[1] reports an observability analysis of 3D SLAM based on IMU sensors within an adaptive Kalman filter. In ref. [9], the authors started from the analysis of ref. [39], investigating the unobservable states and how the observability properties are related to the vehicle motion. All the mentioned works consider standard EKF-SLAM setup in which only the positions of landmarks are estimated. In ref. [14] it is shown that considering the orientation of the features can be beneficial for reducing estimation uncertainty. In this paper we provide an observability analysis for the case in which the full *pose* of the landmarks is estimated, including the interesting case of delayed-state EKF-SLAM.[61] We notice that few other works within the robotics field discussed the use of unit quaternions in 3D SLAM. In the recent paper,[20] the authors investigated a similar problem with application to cooperative navigation of *Micro Aerial Vehicles* (MAV). The authors proposed the use of delayed-state filter expressed in information form to keep track of the trajectories of the agents involved in the navigation process. However, the approach in ref. [20] updates the unit quaternions as four dimensional vectors hence the unit norm is not preserved during filter update. Several authors deal with this issue by normalizing quaternions (i.e., substituting the quaternion $\hat{\boldsymbol{q}}$ with $\hat{\boldsymbol{q}}/\|\hat{\boldsymbol{q}}\|$ after update).[45,52] However, since *normalization* constitutes a further source of errors, Cristofaro *et al.* proposed to use a *projection filter*.[47] In this filter a spurious observation is included in filter update: the robot includes an ideal measurements in the form $\|\hat{\boldsymbol{q}}\| = 1$. It is clear that this heuristic allows to produce estimates that are closer to unit quaternions although this approach unavoidably leads to inconsistency for the introduction of fictitious measurements in the estimation process. On the other hand, our approach does not use heuristics for assuring the unit norm constraint on the rotation quaternions: the norm constraint is intrinsically taken into account in the formulation and does not need to be enforced after the update phase. We remark that our literature review is limited to EKF-based approaches, although other estimation approaches are possible (e.g., using conformal geometric algebra[6,7]) or optimization-based techniques.[18,38]

The last category of related work includes contributions on delayed-state EKF-SLAM. Although this interpretation of pose SLAM directly stems from early work on graph-based SLAM[44] and on Rao-Blackwellized Particle Filters SLAM,[22] incremental approaches were later proposed within Kalman filter framework. Leonard *et al.*[43] investigated the delayed-state approach using sonar data in planar scenarios. Cole and Newman[19] addressed the problem of 3D mapping with laser rangefinder, using a delayed-state architecture based on Euler-angles for rotation representation. Eustice *et al.*[61] provided the first comprehensive formulation, implementing the filter in information form and reporting a useful numerical analysis on the accuracy and the computational effort of their approach. More recently, Ila *et al.*[36] provided a numerical study of the consistency properties of delayed-state SLAM in a planar setup, providing useful heuristics for preventing overconfident covariance estimates by selecting only most informative measurements for loop closing; moreover, the work provided a strategy for selecting the poses to be included in the state vector, so to avoid redundant information in the estimation process. Our contribution enriches the literature on delayed-state EKF-SLAM by

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

proposing a quaternion-based formulation which is proved to be completely observable. Moreover, in the following section we provide a numerical evaluation of the consistency and of the accuracy of the approach, considering also examples of fully 3D problems that may occur in Unmanned Aerial Vehicles navigation.

3. Problem Statement and Motivations

An agent (mobile robot, autonomous vehicle, UAV, etc.) moves in an unstructured non-planar environment with the primary aim of building a consistent representation of the scenario while estimating its own position. The agent is equipped with proprioceptive and exteroceptive sensors. The formers allow to acquire information on the relative displacement of the agent between consecutive sensor measurements (*odometry*); the latter provides measurements of the surrounding scenario.

The odometric information may be provided, for instance, by wheel odometry in planar scenarios,[58] by scan matching procedures,[58] by Inertial Measurement Units,[60] or by vision sensors.[54] All previous sensor data can be easily abstracted in terms of measurement of the relative pose change (roto-translation) between two consecutive sensor measurements. The sensors that are typically used for exteroceptive sensing in 3D robotic navigation are more commonly stereo or monocular cameras[29] or 3D laser scanners.[19,49,62] Such devices produce a large amount of data that are usually processed by an *information synthesis* (preprocessing) block before being included in the SLAM estimation process. The outcome of information synthesis, in common applications, can be roughly classified according to the following categories:

1. *position* of 3D points corresponding to distinguishable point features;[29]
2. *orientation* of objects (e.g., normal to planes, direction of edges) or orientation of topological places (for instance, the orientation of a corridor, which can be easily extracted from laser or camera measurements);
3. *bearing* to landmarks (e.g., azimuth and elevation at which a far landmark is observed);
4. *relative poses* with respect to 3D objects (e.g., other vehicles, distinguishable rigid bodies).

Most part of the 3D SLAM literature considers the first type of information: distinguishable elements (e.g., *visual features*) in the environment are modeled as independent landmarks and filtering techniques (e.g., Extended Kalman Filter) are applied to perform estimation over an augmented state space that includes both robot pose and landmark positions. As commented in the introduction, a naive implementation of such technique quickly leads to huge state vectors (from a pair of stereo images it is possible to extract hundreds of visual features[29]), making prohibitive the use of EKF-SLAM in large-scale problems.

Instead we now want to point out that all the previous categories can be easily abstracted in terms of relative pose measurements. For the last category in the list this abstraction is straightforward. Regarding the first three categories we can notice that (1) positions of points, (2) objects orientations and (3) bearings to far landmarks can be modeled by means of 3D vectors. In particular line directions, normal to planes, and other examples falling in the second and third categories may be represented as *unit vectors* (since they provide only a relative orientation information), whereas positions are described by 3D vectors in the robot frame. Therefore, we can associate, to each observation pose, a collection of 3D vectors that provides a compact representation of robot perception. Then, *registration techniques* (see Appendix D and[12]) can be successfully employed for retrieving the relative transformation (roto-translation) between two sets of 3D vectors, hence to estimate the relative pose between two observation poses. Therefore, all the listed exteroceptive measurements can be treated in terms of relative pose measurements. This motivates our investigation of a 3D SLAM approach based on relative pose measurements. We remark that 3D vectors are also an effective model for other exteroceptive perceptions such as star-trackers, sun sensors, or measurements of the gravity vector by means of accelerometers, enforcing the generality of a relative pose-based SLAM framework.

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

4. EKF-SLAM from Relative Pose Measurements

We now present a discrete-time quaternion-based filter, which is tailored on SLAM with relative pose measurements.

4.1. Filter structure

We will consider a situation in which the state vector at time k includes the robot pose $\boldsymbol{x}_0(k) \doteq [\boldsymbol{p}_0(k)^\top \ \boldsymbol{q}_0(k)^\top]^\top$ and the poses of n *landmarks* $\boldsymbol{x}_j(k) \doteq [\boldsymbol{p}_j(k)^\top \ \boldsymbol{q}_j(k)^\top]^\top$, $j = 1, \ldots, n$. In particular, the generic pose $i = 0, 1, \ldots, n$ includes a Cartesian position $\boldsymbol{p}_i(k) \in \mathbb{R}^4$ (in homogeneous coordinates) and a unit quaternion $\boldsymbol{q}_i(k)$. Hence the overall state vector for SLAM is $\boldsymbol{x}(k) \doteq [\boldsymbol{x}_0^\top(k) \ldots \boldsymbol{x}_n^\top(k)]^\top$. Sometimes we indicate with Ω the set including both robot and landmark poses ($|\Omega| = n + 1$).

As usual in Kalman Filter literature, the index $(k|k-1)$ labels quantities that incorporate information of the process model (e.g., odometry), while the label $(k|k)$ marks quantities that include also the measurements acquired at time k. Therefore, the estimate of the state after the prediction phase (*prior*) is $\hat{\boldsymbol{x}}(k|k-1)$, and the corresponding covariance matrix is $\boldsymbol{P}(k|k-1)$; the estimate of the state after the update phase (*posterior*) is instead $\hat{\boldsymbol{x}}(k|k)$, and the corresponding covariance matrix is $\boldsymbol{P}(k|k)$. The *estimation error* for a generic pose $i \in \Omega$ is further defined as $\tilde{\boldsymbol{x}}_i(k|k) \doteq [\tilde{\boldsymbol{t}}_i(k|k)^\top \ \tilde{\boldsymbol{\theta}}_i(k|k)^\top]^\top \in \mathbb{R}^6$, where $\tilde{\boldsymbol{t}}_i(k|k)^\top$ is the Cartesian error, i.e., $\tilde{\boldsymbol{p}}_i(k|k) = [\tilde{\boldsymbol{t}}_i(k|k)^\top \ 1]^\top$, with $\tilde{\boldsymbol{p}}_i(k|k) = \boldsymbol{p}_i(k) \ominus \hat{\boldsymbol{p}}_i(k|k)$ (difference between actual and estimated position); moreover, given the orientation error expressed (in quaternion form) as $\tilde{\boldsymbol{q}}_i(k|k) = \hat{\boldsymbol{q}}_i(k|k)^{-1}\boldsymbol{q}_i(k)$, the orientation error vector $\tilde{\boldsymbol{\theta}}_i(k|k)^\top$ is such that $\tilde{\boldsymbol{q}}_i(k|k) \approx [\tilde{\boldsymbol{\theta}}_i(k|k)^\top/2 \ \ 1]^\top$, see (A2). Accordingly, the *error state vector* is $\tilde{\boldsymbol{x}}(k|k) \doteq [\tilde{\boldsymbol{x}}_0(k|k)^\top \ldots \tilde{\boldsymbol{x}}_n(k|k)^\top]^\top \in \mathbb{R}^{6(n+1)}$ and the corresponding covariance is $\boldsymbol{P}(k|k) \in \mathbb{R}^{6(n+1)\times 6(n+1)}$. It is worth noticing that the uncertainty associated with a unit quaternion is represented by a 3 by 3 covariance matrix: this is a common choice in aerospace literature, since a 4 by 4 representation for the covariance is singular, due to the unit norm constraint.[41]

4.2. Prediction phase

The prediction phase is aimed at propagating the belief on robot pose according to the odometric information $\boldsymbol{u}(k) = [\boldsymbol{p}_u(k)^\top \boldsymbol{q}_u(k)^\top]^\top$, being $\boldsymbol{p}_u(k)$ and $\boldsymbol{q}_u(k)$ the translation vector and the quaternion describing the pose change between time $k-1$ and time k, respectively. It is worth noticing that $\boldsymbol{u}(k)$ is expressed in the local frame of the robot at time $k-1$. As usual in EKF framework, we assume to have a probabilistic description of the odometry information, i.e., the measured value, namely $\bar{\boldsymbol{u}}(k)$, and a covariance matrix of measurement noise, $\boldsymbol{P}_u(k) \in \mathbb{R}^{6\times 6}$.

A general process model for robot motion, which describes the current robot pose $\boldsymbol{x}_0(k)$, given the previous pose $\boldsymbol{x}_0(k-1)$ and the odometric information $\boldsymbol{u}(k)$ can be written as follows:

$$\begin{cases} \boldsymbol{p}_0(k) = \boldsymbol{p}_0(k-1) \oplus \left(\boldsymbol{q}_0(k-1)\boldsymbol{p}_u(k)\boldsymbol{q}_0(k-1)^{-1}\right) \\ \boldsymbol{q}_0(k) = \boldsymbol{q}_0(k-1)\boldsymbol{q}_u(k) \end{cases} \tag{1}$$

Therefore the *prior* for the pose of the robot is:

$$\begin{cases} \hat{\boldsymbol{p}}_0(k|k-1) = \hat{\boldsymbol{p}}_0(k-1|k-1) \oplus \left(\hat{\boldsymbol{q}}_0(k-1|k-1)\bar{\boldsymbol{p}}_u(k)\hat{\boldsymbol{q}}_0(k-1|k-1)^{-1}\right) \\ \hat{\boldsymbol{q}}_0(k|k-1) = \hat{\boldsymbol{q}}_0(k-1|k-1)\bar{\boldsymbol{q}}_u(k) \end{cases} \tag{2}$$

Assuming that the pose of the landmarks does not change (*static environment*), for a generic landmark j the prediction phase is simply:

$$\begin{cases} \hat{\boldsymbol{p}}_j(k|k-1) = \hat{\boldsymbol{p}}_j(k-1|k-1) \\ \hat{\boldsymbol{q}}_j(k|k-1) = \hat{\boldsymbol{q}}_j(k-1|k-1) \end{cases}$$

In order to devise the covariance matrix after prediction we need to linearize the process model, obtaining the *error state process model*,[60] in which the prediction phase is expressed in terms of

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

estimation errors. The complete derivation can be found in Appendix B, while we now report the final result. The *error state process model for robot motion* is:

$$\tilde{\boldsymbol{x}}_0(k|k-1) = \boldsymbol{F}_k \tilde{\boldsymbol{x}}_0(k-1|k-1) + \boldsymbol{G}_k \tilde{\boldsymbol{u}}(k)$$

where $\tilde{\boldsymbol{x}}_0(k-1|k-1)$ is the *robot posterior error* at time $k-1$, $\tilde{\boldsymbol{x}}_0(k|k-1)$ is the *robot prior error* at time k, $\tilde{\boldsymbol{u}}(k)$ is the odometric error. The matrices $\boldsymbol{F}_k$ and $\boldsymbol{G}_k$ are defined as follows:

$$\boldsymbol{F}_k = \begin{bmatrix} \boldsymbol{I}_3 & -\hat{\boldsymbol{R}}_{k-1}\boldsymbol{S}_{u,k} \\ \boldsymbol{0}_{3,3} & \boldsymbol{R}_{u,k}^\top \end{bmatrix}, \quad \boldsymbol{G}_k = \begin{bmatrix} \hat{\boldsymbol{R}}_{k-1} & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} \tag{3}$$

with $\hat{\boldsymbol{R}}_{k-1} = \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))$, $\boldsymbol{S}_{u,k} = \boldsymbol{S}(\bar{\boldsymbol{p}}_u(k))$ ($\boldsymbol{S}$ is the skew symmetric matrix built as in Eq. (A5), Appendix A), and $\boldsymbol{R}_{u,k} = \boldsymbol{R}(\bar{\boldsymbol{q}}_u(k))$. The full *error state process model* can be then written as:

$$\tilde{\boldsymbol{x}}(k|k-1) = \boldsymbol{\mathcal{F}}_k \tilde{\boldsymbol{x}}(k-1|k-1) + \boldsymbol{\mathcal{G}}_k \tilde{\boldsymbol{u}}(k) \tag{4}$$

with:

$$\boldsymbol{\mathcal{F}}_k = \begin{bmatrix} \boldsymbol{F}_k & \boldsymbol{0}_{6,6} & \dots & \boldsymbol{0}_{6,6} \\ \boldsymbol{0}_{6,6} & \boldsymbol{I}_6 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \boldsymbol{0}_{6,6} \\ \boldsymbol{0}_{6,6} & \dots & \boldsymbol{0}_{6,6} & \boldsymbol{I}_6 \end{bmatrix} \quad \boldsymbol{\mathcal{G}}_k = \begin{bmatrix} \boldsymbol{G}_k \\ \boldsymbol{0}_{6,6} \\ \vdots \\ \boldsymbol{0}_{6,6} \end{bmatrix} \tag{5}$$

The identity blocks in $\boldsymbol{\mathcal{F}}_k$ are connected with the hypothesis of static landmarks. From Eq. (4) it is now easy to write a first-order expression for the covariance of the estimation error of the overall SLAM state after prediction:

$$\boldsymbol{P}(k|k-1) = \boldsymbol{\mathcal{F}}_k \boldsymbol{P}(k-1|k-1)\boldsymbol{\mathcal{F}}_k^\top + \boldsymbol{\mathcal{G}}_k \boldsymbol{P}_u(k)\boldsymbol{\mathcal{G}}_k^\top$$

4.3. Update phase

The update phase allows to include measurements information in the prior $\hat{\boldsymbol{x}}(k|k-1)$. In this context we are interested in the case in which data from exteroceptive sensors are abstracted in the form of relative pose measurements.

Let us define the set of available measurements $\mathcal{M}$, such that, if $j \in \mathcal{M}$, then the robot measured the relative pose with respect to the j-th landmark, namely $\boldsymbol{z}_j(k) \doteq [\boldsymbol{p}_{z_j}(k)^\top \ \boldsymbol{q}_{z_j}(k)^\top]^\top$; then the measurement model can be written as:

$$\begin{cases} \boldsymbol{p}_{z_j}(k) = \boldsymbol{q}_0(k)^{-1}\left(\boldsymbol{p}_j(k) \ominus \boldsymbol{p}_0(k)\right)\boldsymbol{q}_0(k) \\ \boldsymbol{q}_{z_j}(k) = \boldsymbol{q}_0(k)^{-1}\boldsymbol{q}_j(k) \end{cases} \quad \forall j \in \mathcal{M} \tag{6}$$

The previous expression describes the quantities that are observed in the system; however, because of measurement noise the previous expression becomes:

$$\begin{cases} \bar{\boldsymbol{p}}_{z_j}(k) \oplus \tilde{\boldsymbol{p}}_{z_j}(k) = \boldsymbol{q}_0(k)^{-1}\left(\boldsymbol{p}_j(k) \ominus \boldsymbol{p}_0(k)\right)\boldsymbol{q}_0(k) \\ \bar{\boldsymbol{q}}_{z_j}(k)\tilde{\boldsymbol{q}}_{z_j}(k) = \boldsymbol{q}_0(k)^{-1}\boldsymbol{q}_j(k) \end{cases} \quad \forall j \in \mathcal{M} \tag{7}$$

where $\bar{\boldsymbol{z}}_j(k) = [\bar{\boldsymbol{p}}_{z_j}(k)^\top \ \bar{\boldsymbol{q}}_{z_j}(k)^\top]^\top$ is the actual noisy measurement, while $\tilde{\boldsymbol{p}}_{z_j}(k) = [\tilde{\boldsymbol{t}}_{z_j}(k)^\top \ \ 1]^\top$ and $\tilde{\boldsymbol{q}}_{z_j}(k) \approx [\tilde{\boldsymbol{\theta}}_{z_j}(k)^\top/2 \ \ 1]^\top$ are random variables describing measurement error. As in the prediction phase, a probabilistic description of the measurement error $\tilde{\boldsymbol{z}}_j = [\tilde{\boldsymbol{t}}_{z_j}(k)^\top \ \ \tilde{\boldsymbol{\theta}}_{z_j}(k)^\top]^\top$ is supposed to be available, in the form of the covariance matrix $\boldsymbol{P}_{z_j}(k) \in \mathbb{R}^{6\times 6}$, $j \in \mathcal{M}$.

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

To devise the equations for the EKF-SLAM update we need to compute a first-order expression for the measurement model, in terms of the involved errors. First of all, we recall the definition of *innovation for the j-th measurement*, that represents the mismatch between the expected and the actual measurement. Let us consider the j-th measurement in Eq. (6) and substitute the estimated values for each variable: the *expected Cartesian measurement* is $\hat{\boldsymbol{p}}_{z_j}(k) \doteq \hat{\boldsymbol{q}}_0(k|k-1)^{-1}\left(\hat{\boldsymbol{p}}_j(k|k-1) \ominus \hat{\boldsymbol{p}}_0(k|k-1)\right)\hat{\boldsymbol{q}}_0(k|k-1)$; the *expected orientation measurement* is instead $\hat{\boldsymbol{q}}_{z_j}(k|k-1) \doteq \hat{\boldsymbol{q}}_0(k|k-1)^{-1}\hat{\boldsymbol{q}}_j(k|k-1)$. Then, the *innovation for the Cartesian variables* is $\tilde{\boldsymbol{p}}_{r_j}(k) \doteq \hat{\boldsymbol{p}}_{z_j}(k|k-1) \ominus \bar{\boldsymbol{p}}_{z_j}(k) = [\tilde{\boldsymbol{t}}_{r_j}^\top \;\; 1]^\top$ and the *innovation for the orientation variables* is $\tilde{\boldsymbol{q}}_{r_j}(k) \doteq \hat{\boldsymbol{q}}_{z_j}(k|k-1)^{-1}\bar{\boldsymbol{q}}_{z_j}(k) \approx [\tilde{\boldsymbol{\theta}}_{r_j}^\top/2 \;\; 1]^\top$. Finally we can write the *innovation for the j-th measurement* as $\tilde{\boldsymbol{r}}_j(k) = [\tilde{\boldsymbol{t}}_{r_j}(k)^\top \;\; \tilde{\boldsymbol{\theta}}_{r_j}(k)^\top]^\top \in \mathbb{R}^6$. Stacking the innovation vectors of all the measurements in a single vector we obtain the overall innovation $\tilde{\boldsymbol{r}}(k)$.

According to the derivation reported in Appendix C, a first-order approximation of the *innovation for the j-th measurement* $\tilde{\boldsymbol{r}}_j$ can be written as:

$$\tilde{\boldsymbol{r}}_j(k) \approx \boldsymbol{H}_j\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{z}}_j \tag{8}$$

where:

$$\begin{aligned}
\boldsymbol{H}_j &= \left[\begin{bmatrix} -\hat{\boldsymbol{R}}_k^\top & \hat{\boldsymbol{S}}_k \\ \boldsymbol{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_k}^\top \end{bmatrix} \boldsymbol{0}_{6,6} \cdots \boldsymbol{0}_{6,6} \begin{bmatrix} \hat{\boldsymbol{R}}_k^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} \boldsymbol{0}_{6,6} \cdots \boldsymbol{0}_{6,6}\right] \\
\tilde{\boldsymbol{x}} &= \begin{bmatrix} \tilde{\boldsymbol{t}}_0(k|k-1) \\ \tilde{\boldsymbol{\theta}}_0(k|k-1) \\ \vdots \\ \tilde{\boldsymbol{t}}_j(k|k-1) \\ \tilde{\boldsymbol{\theta}}_j(k|k-1) \\ \vdots \end{bmatrix} \\
\tilde{\boldsymbol{z}}_j &= \begin{bmatrix} \tilde{\boldsymbol{t}}_{z_j}(k) \\ \tilde{\boldsymbol{\theta}}_{z_j}(k) \end{bmatrix} \\
\hat{\boldsymbol{R}}_k &= \boldsymbol{R}\left(\hat{\boldsymbol{q}}_0(k|k-1)\right) \\
\hat{\boldsymbol{R}}_{\Delta_k} &= \boldsymbol{R}\left(\hat{\boldsymbol{q}}_0(k|k-1)^{-1}\hat{\boldsymbol{q}}_j(k|k-1)\right) \\
\hat{\boldsymbol{S}}_k &= \boldsymbol{S}\left(\boldsymbol{R}^\top\left(\hat{\boldsymbol{q}}_0(k|k-1)\right)\left(\hat{\boldsymbol{t}}_j(k|k-1) - \hat{\boldsymbol{t}}_0(k|k-1)\right)\right)
\end{aligned} \tag{9}$$

Stacking the matrices $\boldsymbol{H}_j$ in a single matrix $\boldsymbol{\mathcal{H}}_k$ we obtain an expression for the overall *innovation* in function of the estimation and measurement errors:

$$\tilde{\boldsymbol{r}}(k) \approx \boldsymbol{\mathcal{H}}_k\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{z}} \tag{10}$$

From Eq. (10), it is now possible to derive *the covariance matrix for the innovation*:

$$\boldsymbol{P}_r(k) = \boldsymbol{\mathcal{H}}_k\boldsymbol{P}(k|k-1)\boldsymbol{\mathcal{H}}_k^\top + \boldsymbol{P}_z(k)$$

with $\boldsymbol{P}_z(k)$ being the covariance matrix of all the available measurements $\boldsymbol{z}(k)$, i.e., $\boldsymbol{P}_z(k) = \text{diag}(\boldsymbol{P}_{z_1}(k), \boldsymbol{P}_{z_2}(k), \ldots, \boldsymbol{P}_{z_{|\mathcal{M}|}}(k))$. Hence, the *Kalman gain* can be computed as:[5]

$$\boldsymbol{\mathcal{K}}_k = \boldsymbol{P}(k|k-1)\boldsymbol{\mathcal{H}}_k^\top\boldsymbol{P}_r(k)^{-1} \tag{11}$$

Given the innovation $\tilde{\boldsymbol{r}}(k)$ and the gain (11) the correction term for the state can be written as $\boldsymbol{c}(k) = \boldsymbol{\mathcal{K}}_k\tilde{\boldsymbol{r}}(k)$. Now, if we separate the Cartesian and the orientation variables in $\boldsymbol{c}(k)$ for the generic

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

pose $i \in \Omega$, namely $\tilde{\boldsymbol{t}}_{c_i}(k)$ and $\tilde{\boldsymbol{\theta}}_{c_i}(k)$ and we obtain the homogeneous coordinates corresponding to $\tilde{\boldsymbol{t}}_{c_i}(k)$, i.e., $\boldsymbol{p}_{c_i}(k)$, and the quaternion corresponding to $\tilde{\boldsymbol{\theta}}_{c_i}(k)$, i.e., $\boldsymbol{q}_{c_i}(k) = \Phi\big(\tilde{\boldsymbol{\theta}}_{c_i}(k)\big)$ ($\Phi(\cdot)$ is the exponential map that maps a rotation vector in $\mathbb{R}^3$ to the corresponding unit quaternion, see Appendix A), the update equations for the state vector can be written as:

$$\begin{cases} \hat{\boldsymbol{p}}_i(k|k) = \hat{\boldsymbol{p}}_i(k|k-1) \oplus \boldsymbol{p}_{c_i}(k) \\ \hat{\boldsymbol{q}}_i(k|k) = \hat{\boldsymbol{q}}_i(k|k-1)\boldsymbol{q}_{c_i}(k) \end{cases} \quad i = 0, \ldots, n$$

Moreover the covariance matrix for the posterior is:

$$\boldsymbol{P}(k|k) = \big(\boldsymbol{I}_{6(n+1)} - \mathcal{K}_k \mathcal{H}_k\big)\, \boldsymbol{P}(k|k-1)$$

In the following section we use the expressions we derived so far to discuss the observability properties of the proposed approach.

5. Observability Analysis

Without loss of generality we now consider a delayed-state architecture, in which the landmarks included in the state vectors correspond to past observation poses assumed by the robot. The original system is clearly nonlinear and, in this context, we employ the machinery introduced in ref. [30] (and applied in a planar SLAM problem in ref. [34]) for evaluating the observability properties of the piecewise linear model used in EKF-SLAM. According to the derivation of Section 4.2 and 4.3 the system can be modeled through the following time-variant linear system:

$$\begin{cases} \tilde{\boldsymbol{x}}(k) = \mathcal{F}_k \tilde{\boldsymbol{x}}(k-1) + \mathcal{G}_k \tilde{\boldsymbol{u}}(k) \\ \tilde{\boldsymbol{r}}(k) = \mathcal{H}_k \tilde{\boldsymbol{x}}(k) - \tilde{\boldsymbol{z}}(k) \end{cases} \tag{12}$$

The subscript k of systems matrixes ($\mathcal{F}_k$, $\mathcal{G}_k$, $\mathcal{H}_k$) remarks their dependence on time, and for each discrete-time segment, the system can be approximated by the previous linear model. Without loss of generality let us assume that, at each pose, the robot initializes a new 3D landmark with the current observation pose; moreover the robot, at time k, is able to measure the relative pose with respect to the landmark initialized at time $k-1$ (e.g., using the vector registration algorithm provided in Appendix D). Under the assumptions considered so far, we can write $\mathcal{F}_k$ as:

$$\mathcal{F}_k = \begin{bmatrix} \begin{bmatrix} \boldsymbol{I}_3 & -\hat{\boldsymbol{R}}_{k-1}\boldsymbol{S}_{u,k} \\ \boldsymbol{0}_{3,3} & \boldsymbol{R}_{u,k}^\top \end{bmatrix} & \boldsymbol{0}_{6,6} & \cdots & \boldsymbol{0}_{6,6} \\ \boldsymbol{0}_{6,6} & [\boldsymbol{I}_6] & & \boldsymbol{0}_{6,6} \\ \vdots & & \ddots & \\ \boldsymbol{0}_{6,6} & & & [\boldsymbol{I}_6] \end{bmatrix} \tag{13}$$

where the upper left block contains the linearized motion model of the robot, used in the prediction phase, and the lower-right identity blocks contains the landmark process model. Similarly, we can rewrite the measurement Jacobian matrix as:

$$\mathcal{H}_k = \begin{bmatrix} \begin{bmatrix} -\hat{\boldsymbol{R}}_k^\top & \hat{\boldsymbol{S}}_k \\ \boldsymbol{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_k}^\top \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_k^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} & \boldsymbol{0}_{6,6} & \cdots & \boldsymbol{0}_{6,6} \end{bmatrix} \tag{14}$$

where the previous expression is obtained from (8), assuming that the robot observes the pose of the landmark initialized at time $k-1$. Now we need to introduce few concepts for investigating the observability properties of the system at hand. According to the results reported in ref. [9, 30] we can study the local observability properties of the system with linear algebra tools. In particular the system is said to be *locally (instantaneously) observable* at time k if the matrix

$$\mathcal{O}_k = \begin{bmatrix} (\mathcal{H}_k)^\top & (\mathcal{H}_k\mathcal{F}_k)^\top & (\mathcal{H}_k\mathcal{F}_k^2)^\top & \cdots & (\mathcal{H}_k\mathcal{F}_k^{\eta-1})^\top \end{bmatrix}^\top \tag{15}$$

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

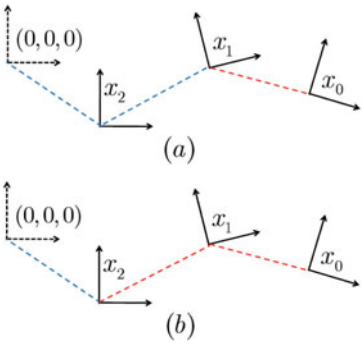

Fig. 1. Simplified (planar) example of delayed-state SLAM with 2 landmarks. The landmark poses, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ (respectively initialized at time $k = 1$ and $k = 0$), are denoted with two reference frames, whereas $\boldsymbol{x}_0$ is the pose of the robot at time $k = 2$. The plot also contains the absolute frame with respect to which the poses are estimated, denoted with (0, 0, 0). The sub-figure (a) is referred to the case in which the observability analysis only considers the last measurement (in red) neglecting the other relative pose measurements (in blue). The robot pose is not observable since it is not possible to infer the absolute pose $\boldsymbol{x}_0$ with respect to the frame (0, 0, 0) using only the last relative measurement. In (b) instead, both segments are considered (in red): also in this case the absolute poses are not observable because we still need to introduce the relationship (*grounding*) between one node and the absolute frame (0, 0, 0) in which localization is performed.

is full rank (i.e., the rank is equal to the dimension of the state space, namely η). The matrix $\mathcal{O}_k$ is usually referred to as *local observability matrix* (LOM). Roughly speaking the concept of local observability is connected to the possibility of instantaneously distinguishing the state $\tilde{\boldsymbol{x}}_k$ from other vectors in the state space using the current observation. Tailoring this concept to the problem at hand, the piecewise linear system describing our SLAM problem is instantaneously observable at time k if the measurements acquired at time k allows to estimate the absolute poses of robot and landmarks. For the study of local observability let us write the LOM in explicit form for the system with $\mathcal{F}_k$ and $\mathcal{H}_k$ described as in (13) and (14), respectively:

$$\mathcal{O}_k = \begin{bmatrix} \begin{bmatrix} -\hat{\boldsymbol{R}}_k^\top & \hat{\boldsymbol{S}}_k \\ \boldsymbol{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_k}^\top \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_k^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} & \boldsymbol{0}_{6,6} & \dots & \boldsymbol{0}_{6,6} \\ \begin{bmatrix} -\hat{\boldsymbol{R}}_k^\top & \boldsymbol{M}_{k,k-1}^1 \\ \boldsymbol{0}_{3,3} & \boldsymbol{N}_{k,k-1}^1 \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_k^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} & \boldsymbol{0}_{6,6} & \dots & \boldsymbol{0}_{6,6} \\ \vdots & \vdots & & \vdots & \\ \begin{bmatrix} -\hat{\boldsymbol{R}}_k^\top & \boldsymbol{M}_{k,k-1}^{\eta-1} \\ \boldsymbol{0}_{3,3} & \boldsymbol{N}_{k,k-1}^{\eta-1} \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_k^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} & \boldsymbol{0}_{6,6} & \dots & \boldsymbol{0}_{6,6} \end{bmatrix}$$

where $\boldsymbol{M}_{k,k-1}^i = \hat{\boldsymbol{R}}_k^\top \hat{\boldsymbol{R}}_{k-1} \boldsymbol{S}_{u,k} \sum_{j=0}^{i-1} \left(\boldsymbol{R}_{u,k}^\top\right)^j + \hat{\boldsymbol{S}}_k \left(\boldsymbol{R}_{u,k}^\top\right)^i$ and $\boldsymbol{N}_{k,k-1}^i = -\hat{\boldsymbol{R}}_{\Delta_k}^\top \left(\boldsymbol{R}_{u,k}^\top\right)^i$. In this case the (column) rank of the LOM can be easily seen to be less than η, since the last columns of the matrix contain only zeros. The case under consideration is depicted in Fig. 1(a): it is clear that the absolute pose of the robot is not observable using only the current observation and for this reason the LOM is not full rank.

We can now introduce a milder definition of observability that allows to take into consideration all the measurements collected in a given interval of time, say $[1, k]$: in this case we are interested in studying if all the observations acquired by the robot since the initial time are sufficient for the estimation of the absolute poses included in the state vector. For studying the observability over the

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

interval $[1, k]$, we introduce the *total observability matrix* (TOM):

$$\mathcal{O}_{[1:k]} = \begin{bmatrix} \mathcal{O}_1 \\ \mathcal{O}_2 \mathcal{F}_1^{\eta-1} \\ \mathcal{O}_3 \mathcal{F}_2^{\eta-1} \mathcal{F}_1^{\eta-1} \\ \vdots \\ \mathcal{O}_k \mathcal{F}_{k-1}^{\eta-1} \mathcal{F}_{k-2}^{\eta-1} \dots \mathcal{F}_1^{\eta-1} \end{bmatrix} \tag{16}$$

Then the following result holds true.[30]

Proposition 1. The discrete piecewise linear system defined in (12) is *completely observable* if and only if the total observability matrix (16) has rank equal to η (dimension of the state space). □

For sake of simplicity let us consider a scenario with two landmarks, in which the size of the error state vector is $\eta = 6(n + 1) = 18$ (6 variables for each landmark plus 6 for the current robot pose). This simplification is made without loss of generality and the general case can be easily derived from the proposed analysis. In this simplified example there are only two time segments: the first corresponds to the time needed for traveling from the landmark 1 to landmark 2, the second corresponding to the time for reaching the current pose, see Fig. 1(b). Therefore, in this case, the TOM assumes the form $\mathcal{O}_{[1:2]} = [(\mathcal{O}_1)^\top \ (\mathcal{O}_2 \mathcal{F}_1^{\eta-1})^\top]^\top$, since $k = 2$. The matrix $\mathcal{O}_2$ corresponds to (15) with $k = 2$, whereas the matrix $\mathcal{F}_1$ has the following form:

$$\mathcal{F}_1 = \begin{bmatrix} [\boldsymbol{I}_6] & \mathbf{0}_{6,6} & \mathbf{0}_{6,6} \\ \mathbf{0}_{6,6} & \begin{bmatrix} \boldsymbol{I}_3 & -\hat{\boldsymbol{R}}_0 S_{u,1} \\ \mathbf{0}_{3,3} & \boldsymbol{R}_{u,1}^\top \end{bmatrix} & \mathbf{0}_{6,6} \\ \mathbf{0}_{6,6} & \mathbf{0}_{6,6} & [\boldsymbol{I}_6] \end{bmatrix} \tag{17}$$

In order to avoid the complexity of the landmark initialization process, i.e., the state augmentation deriving from the inclusion of a new pose in the state vector, we are considering a state vector of constant dimension ($\eta = 18$ in our case) in which they are already included all the landmarks that will be initialized by the robot. In practice the augmentation will take place after time $k = 1$, hence the matrix $\mathcal{F}_1$ of a real implementation will be only constituted only by the 12×12 lower-right block of (17), according to the general structure reported in (13). For building the TOM we need to compute $\mathcal{O}_1$ and $\mathcal{O}_2 \mathcal{F}_1^{\eta-1}$. The former can be easily obtained from (15) substituting $k = 1$ and with a suitable ordering of the columns:

$$\mathcal{O}_1 = \begin{bmatrix} \mathbf{0}_{6,6} & \begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \hat{\boldsymbol{S}}_1 \\ \mathbf{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_1}^\top \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} \\ \mathbf{0}_{6,6} & \begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \boldsymbol{M}_{1,0}^1 \\ \mathbf{0}_{3,3} & \boldsymbol{N}_{1,0}^1 \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} \\ \vdots & \vdots & \vdots \\ \mathbf{0}_{6,6} & \begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \boldsymbol{M}_{1,0}^{\eta-1} \\ \mathbf{0}_{3,3} & \boldsymbol{N}_{1,0}^{\eta-1} \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} \end{bmatrix} \tag{18}$$

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

The matrix $\mathcal{O}_2\mathcal{F}_1^{\eta-1}$ is instead:

$$\mathcal{O}_2\mathcal{F}_1^{\eta-1} = \begin{bmatrix} \begin{bmatrix} -\hat{\boldsymbol{R}}_2^\top & \hat{\boldsymbol{S}}_2 \\ \boldsymbol{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_2}^\top \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_2^\top & \boldsymbol{T}_{2,0} \\ \boldsymbol{0}_{3,3} & \boldsymbol{Q}_1 \end{bmatrix} & \boldsymbol{0}_{6,6} \\ \begin{bmatrix} -\hat{\boldsymbol{R}}_2^\top & \boldsymbol{M}_{2,1}^1 \\ \boldsymbol{0}_{3,3} & \boldsymbol{N}_{2,1}^1 \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_2^\top & \boldsymbol{T}_{2,0} \\ \boldsymbol{0}_{3,3} & \boldsymbol{Q}_1 \end{bmatrix} & \boldsymbol{0}_{6,6} \\ \vdots & \vdots & \vdots \\ \begin{bmatrix} -\hat{\boldsymbol{R}}_2^\top & \boldsymbol{M}_{2,1}^{\eta-1} \\ \boldsymbol{0}_{3,3} & \boldsymbol{N}_{2,1}^{\eta-1} \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_2^\top & \boldsymbol{T}_{2,0} \\ \boldsymbol{0}_{3,3} & \boldsymbol{Q}_1 \end{bmatrix} & \boldsymbol{0}_{6,6} \end{bmatrix} \tag{19}$$

with $\boldsymbol{T}_{2,0} = -\hat{\boldsymbol{R}}_2^\top \hat{\boldsymbol{R}}_0 \boldsymbol{S}_{u,1} \sum_{j=0}^{\eta-2} (\boldsymbol{R}_{u,1}^\top)^j$ and $\boldsymbol{Q}_1 = (\boldsymbol{R}_{u,1}^\top)^{\eta-1}$. Stacking the matrices (18) and (19) in the TOM we obtain:

$$\mathcal{O} = \begin{bmatrix} \overbrace{\phantom{\boldsymbol{0}_{6,6}}}^{1} & \overbrace{\phantom{\boldsymbol{0}_{6,6}}}^{2} & \overbrace{\phantom{\begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \hat{\boldsymbol{S}}_1 \\ \boldsymbol{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_1}^\top \end{bmatrix}}}^{3} & \overbrace{\phantom{\begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \hat{\boldsymbol{S}}_1 \\ \boldsymbol{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_1}^\top \end{bmatrix}}}^{4} & \overbrace{\phantom{\begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix}}}^{5} & \overbrace{\phantom{\begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix}}}^{6} \\ \boldsymbol{0}_{6,6} & & \begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \hat{\boldsymbol{S}}_1 \\ \boldsymbol{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_1}^\top \end{bmatrix} & & \begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} & \\ \boldsymbol{0}_{6,6} & & \begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \boldsymbol{M}_{1,0}^1 \\ \boldsymbol{0}_{3,3} & \boldsymbol{N}_{1,0}^1 \end{bmatrix} & & \begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} & \\ \vdots & & \vdots & & \vdots & \\ \boldsymbol{0}_{6,6} & & \begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \boldsymbol{M}_{1,0}^{\eta-1} \\ \boldsymbol{0}_{3,3} & \boldsymbol{N}_{1,0}^{\eta-1} \end{bmatrix} & & \begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} & \\ \begin{bmatrix} -\hat{\boldsymbol{R}}_2^\top & \hat{\boldsymbol{S}}_2 \\ \boldsymbol{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_2}^\top \end{bmatrix} & & \begin{bmatrix} \hat{\boldsymbol{R}}_2^\top & \boldsymbol{T}_{2,0} \\ \boldsymbol{0}_{3,3} & \boldsymbol{Q}_{2,0} \end{bmatrix} & & \boldsymbol{0}_{6,6} & \\ \begin{bmatrix} -\hat{\boldsymbol{R}}_2^\top & \boldsymbol{M}_{2,1}^1 \\ \boldsymbol{0}_{3,3} & \boldsymbol{N}_{2,1}^1 \end{bmatrix} & & \begin{bmatrix} \hat{\boldsymbol{R}}_2^\top & \boldsymbol{T}_{2,0} \\ \boldsymbol{0}_{3,3} & \boldsymbol{Q}_{2,0} \end{bmatrix} & & \boldsymbol{0}_{6,6} & \\ \vdots & & \vdots & & \vdots & \\ \begin{bmatrix} -\hat{\boldsymbol{R}}_2^\top & \boldsymbol{M}_{2,1}^{\eta-1} \\ \boldsymbol{0}_{3,3} & \boldsymbol{N}_{2,1}^{\eta-1} \end{bmatrix} & & \begin{bmatrix} \hat{\boldsymbol{R}}_2^\top & \boldsymbol{T}_{2,0} \\ \boldsymbol{0}_{3,3} & \boldsymbol{Q}_{2,0} \end{bmatrix} & & \boldsymbol{0}_{6,6} & \end{bmatrix}$$

We can now study the observability property of our model by analyzing the column rank of $\mathcal{O}$. We can identify six block-columns, each one containing three columns. It is then possible to see that the third block column can be obtained as linear combination of the first and the fifth block-columns, hence the matrix is rank deficient, and the system is not completely observable. This is quite an intuitive result since we do not expect to be able to estimate absolute poses using only relative measurements: without imposing a reference frame only the relative poses can be observable. In delayed-state EKF-SLAM it is common to assume the initial pose of the robot to be the reference frame for the mapping process. This assumption will be lately referred to as *grounding*, and the pose that is assumed to be known will be called *anchor*. The observability study can include the grounding process in two different ways: (i) including an absolute measurement of the initial pose, or (ii) marginalizing the sub-set of the state vector corresponding to the initial pose (i.e., the initial pose is assumed to be known and needs not be estimated by the EKF). We follow the first way, whereas the second approach can be easily seen to provide analogous results. An absolute measurement of pose at initial time corresponds to the following measurement matrix:

$$\mathcal{H}_0 = \begin{bmatrix} \boldsymbol{0}_{6,6} & \boldsymbol{0}_{6,6} & \boldsymbol{I}_{6,6} \end{bmatrix}$$

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

Considering this measurement, the TOM becomes:

$$\bar{\mathcal{O}} = \begin{bmatrix} \mathbf{0}_{6,6} & \mathbf{0}_{6,6} & \boldsymbol{I}_{6,6} \\ \vdots & \vdots & \vdots \\ \mathbf{0}_{6,6} & \begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \hat{\boldsymbol{S}}_1 \\ \mathbf{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_1}^\top \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} \\ \mathbf{0}_{6,6} & \begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \boldsymbol{M}_{1,0}^1 \\ \mathbf{0}_{3,3} & \boldsymbol{N}_{1,0}^1 \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} \\ \vdots & \vdots & \vdots \\ \mathbf{0}_{6,6} & \begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \boldsymbol{M}_{1,0}^{\eta-1} \\ \mathbf{0}_{3,3} & \boldsymbol{N}_{1,0}^{\eta-1} \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} \\ \begin{bmatrix} -\hat{\boldsymbol{R}}_2^\top & \hat{\boldsymbol{S}}_2 \\ \mathbf{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_2}^\top \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_2^\top & \boldsymbol{T}_{2,0} \\ \mathbf{0}_{3,3} & \boldsymbol{Q}_{2,0} \end{bmatrix} & \mathbf{0}_{6,6} \\ \begin{bmatrix} -\hat{\boldsymbol{R}}_2^\top & \boldsymbol{M}_{2,1}^1 \\ \mathbf{0}_{3,3} & \boldsymbol{N}_{2,1}^1 \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_2^\top & \boldsymbol{T}_{2,0} \\ \mathbf{0}_{3,3} & \boldsymbol{Q}_{2,0} \end{bmatrix} & \mathbf{0}_{6,6} \\ \vdots & \vdots & \vdots \\ \begin{bmatrix} -\hat{\boldsymbol{R}}_2^\top & \boldsymbol{M}_{2,1}^{\eta-1} \\ \mathbf{0}_{3,3} & \boldsymbol{N}_{2,1}^{\eta-1} \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_2^\top & \boldsymbol{T}_{2,0} \\ \mathbf{0}_{3,3} & \boldsymbol{Q}_{2,0} \end{bmatrix} & \mathbf{0}_{6,6} \end{bmatrix}$$

In order to study the rank of the previous matrix let us consider the following matrix $\tilde{\mathcal{O}}$, obtained by selecting η rows from $\bar{\mathcal{O}}$:

$$\tilde{\mathcal{O}} = \begin{bmatrix} \begin{bmatrix} -\hat{\boldsymbol{R}}_2^\top & \hat{\boldsymbol{S}}_2 \\ \mathbf{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_2}^\top \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_2^\top & \boldsymbol{T}_{2,0} \\ \mathbf{0}_{3,3} & \boldsymbol{Q}_{2,0} \end{bmatrix} & \mathbf{0}_{6,6} \\ \mathbf{0}_{6,6} & \begin{bmatrix} -\hat{\boldsymbol{R}}_1^\top & \hat{\boldsymbol{S}}_1 \\ \mathbf{0}_{3,3} & -\hat{\boldsymbol{R}}_{\Delta_1}^\top \end{bmatrix} & \begin{bmatrix} \hat{\boldsymbol{R}}_1^\top & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} \\ \mathbf{0}_{6,6} & \mathbf{0}_{6,6} & \begin{bmatrix} \boldsymbol{I}_{6,6} \end{bmatrix} \end{bmatrix}$$

Because of the triangular block-structure of the matrix $\tilde{\mathcal{O}}$, the determinant of $\tilde{\mathcal{O}}$ can be computed by multiplying the determinants of the blocks composing the main diagonal,[33] i.e., $\det\tilde{\mathcal{O}} = \det(-\hat{\boldsymbol{R}}_2^\top)\det(-\hat{\boldsymbol{R}}_{\Delta_2}^\top)\det(-\hat{\boldsymbol{R}}_1^\top)\det(-\hat{\boldsymbol{R}}_{\Delta_1}^\top)\det(\boldsymbol{I}_{6,6})$. Since the involved matrices are rotation or identity matrices we obtain $\det\tilde{\mathcal{O}} = 1$. The determinant is different from zero, then the matrix $\det\tilde{\mathcal{O}}$ is full rank; therefore, if a subset of η rows in $\mathcal{O}$ are already linearly independent, also the rank of $\mathcal{O}$ is η, and the system is completely observable.

In conclusion, we demonstrated that, assuming a pose to be known, allows to prove that the absolute poses included in the state vector in SLAM are completely observable from relative pose measurements over the interval $[1, k]$. It is intuitive that the lack of local observability stems from the impossibility of exploiting absolute information from relative measurements, hence the system becomes locally observable at time k only if the measurement matrix $\mathcal{H}_k$ includes at least one measurement referred to the anchor pose. This aspect will be further underlined in Section 6.1.

6. Experiments and Discussion

6.1. Numerical experiments

The numerical analysis is mainly concerned with the study of *consistency* of the SLAM approach, i.e., we evaluate if the covariance matrices computed with the proposed filter are compatible with the corresponding estimation errors.[5] Consistency is crucial in Probabilistic Robotics since a robot that is unable to quantify the uncertainty on the variables of interest (poses of the robot or of 3D landmarks in the considered setup) acts in a *misleading optimism* which may lead to catastrophic

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

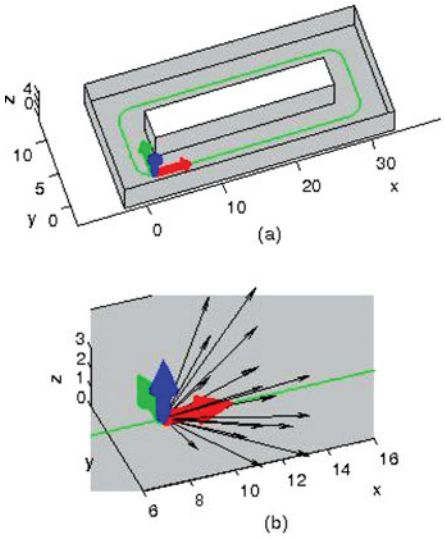

Fig. 2. (a) Ground truth trajectory (green line) and starting pose of the robot moving in a loop-shaped corridor (axis scale in meters); (b) Example of 3D vectors corresponding to a sensor measurement: the robot pose is denoted with three arrows (red-green-blue), whereas the 3D vectors are shown in black. In vision-based applications one may consider vectors describing the position of distinguishable visual features (e.g., SURF).

failure. In SLAM problems, consistency directly affects estimation results: if the robot underestimates the uncertainty on its pose it risks to discard potential loop closings when performing data association with previously observed landmarks: this is undesirable since loop closings offer the robot the possibility to drastically reduce the localization error. On the other hand, an overestimation of the uncertainty leads to false positives in data association that cause the failure of the mapping process. In this section we evaluate the consistency of the proposed approach, discussing numerical evidence. To the best of our knowledge this is the first work performing such analysis in a 3D-SLAM setup.

Let us start by considering the case in which a robot travels the loop in Fig. 2(a) three times. The robot is denoted by a local reference frame (*red-green-blue* axes) which uniquely identifies its pose: the origin of the frame denotes robot position whereas the axes define its orientation (`red` = x-axis, aligned with the direction of robot motion; `blue` = z-axis, perpendicular to the robot platform and pointing upwards; `green` = y-axis, completing the reference frame). When traveling in the environment, the robot implements the delayed-state EKF-SLAM approach discussed in this paper. At time $k = 0$, the robot initializes a landmark corresponding to its own pose and sets this landmark at the origin of the reference frame; the initial covariance matrix of robot pose is set to $\boldsymbol{P}(0|0) = \mathbf{0}_{6,6}$ (the choice of the reference frame is arbitrary and no source of uncertainty influences it). Moreover, the robot associates to the initialized landmark a set of vectors corresponding to the exteroceptive measurement taken at time $k = 0$, e.g., Fig. 2(b).

At time $k = 1$ the robot moves and acquires a 3D odometric measurement $\bar{\boldsymbol{u}}(1)$, describing the relative displacement between its pose at time $k = 0$ and time $k = 1$ (Section 4.2); the covariance matrix describing the noise on odometric data is supposed to be known and, for the numerical experiments, we use the following *process noise covariance* $\boldsymbol{P}_u(k) = \text{diag}(\sigma^2_{u,t}\boldsymbol{I}_3, \sigma^2_{u,\theta}\boldsymbol{I}_3)$, where $\sigma^2_{u,t}$ and $\sigma^2_{u,\theta}$ are the *Cartesian variance* and *orientation variance* of the odometric data, respectively. For sake of simplicity, the nominal trajectory is planar, but such information is not used in the experiments and all the errors are referred to a 3D setup. At time $k = 1$ the robot also acquires a second exteroceptive measurement. As before, this measurement is described by a collection of 3D vectors, each one characterized by a *unique descriptor*. According to the algorithm in Appendix D, the robot can now obtain relative pose measurements with respect to the previous landmark, by computing the roto-translation between the current set of 3D vectors and the corresponding measurements acquired at time $k = 0$. In Section 6.2 we describe how to extract such information in real-world examples. Now, instead, let us simply assume that we can directly measure the relative pose between the current robot position and a given landmark in the state vector, when at least m 3D vectors

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

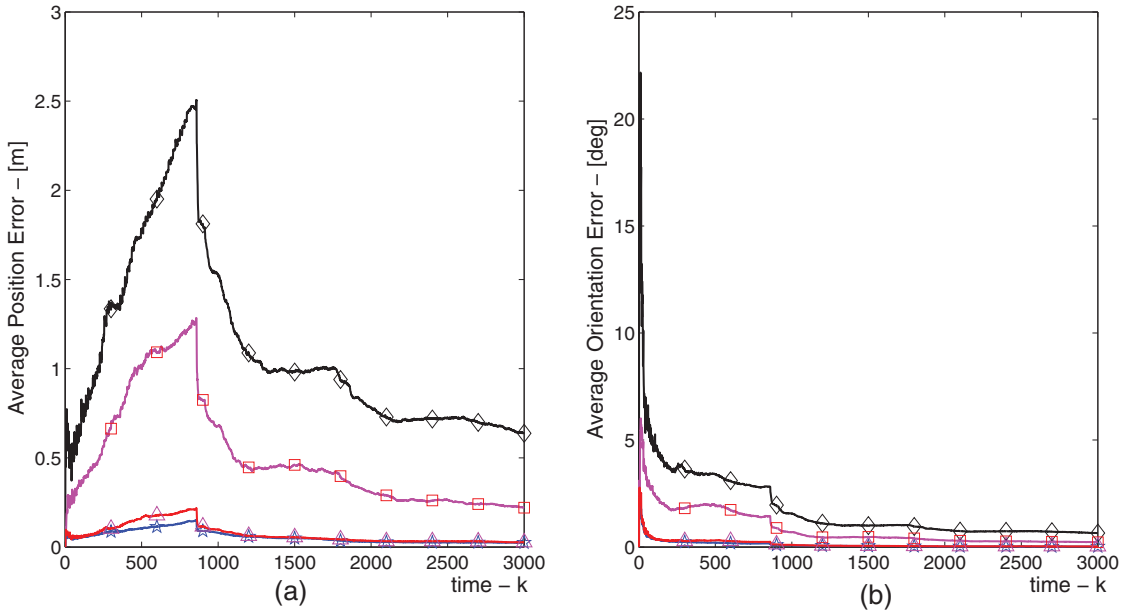

Fig. 3. Numerical results: Average position and orientation errors for different values of uncertainty. Blue (★ marker): $\{\sigma_{u,t} = 0.1$ m, $\sigma_{u,\theta} = 1^o$, $\sigma_{z,t} = 0.1$ m, $\sigma_{z,\theta} = 1^o\}$; Red (△ marker): $\{\sigma_{u,t} = 1$ m, $\sigma_{u,\theta} = 10^o$, $\sigma_{z,t} = 0.1$ m, $\sigma_{z,\theta} = 1^o\}$; Magenta (□ marker): $\{\sigma_{u,t} = 0.1$ m, $\sigma_{u,\theta} = 1^o$, $\sigma_{z,t} = 1$ m, $\sigma_{z,\theta} = 10^o\}$; Black (◇ marker): $\{\sigma_{u,t} = 1$ m, $\sigma_{u,\theta} = 10^o$, $\sigma_{z,t} = 1$ m, $\sigma_{z,\theta} = 10^o\}$.

were observed from both poses (we set $m = 50$). Moreover we assume that the covariance matrix of the measurements is $\boldsymbol{P}_z(k) = \text{diag}(\sigma_{z,t}^2 \boldsymbol{I}_3, \sigma_{z,\theta}^2 \boldsymbol{I}_3)$, where $\sigma_{z,t}^2$ and $\sigma_{z,\theta}^2$ are the *Cartesian variance* and *orientation variance* of the measurements, respectively. The estimation process is iterated as the robot moves; in particular, every 20 time steps, the robot checks if the current observation contains a sufficient number of known 3D vectors, otherwise it initializes a new landmark. When adding a new landmark to the $n + 1$ poses that are already in the state vector, the covariance matrix augmentation works as follows:

$$\boldsymbol{P}(k|k) = \begin{bmatrix} \boldsymbol{I}_6 & \boldsymbol{0}_{6,6n} \\ \boldsymbol{I}_6 & \boldsymbol{0}_{6,6n} \\ \boldsymbol{0}_{6n,6} & \boldsymbol{I}_{6n} \end{bmatrix} \boldsymbol{P}^-(k|k) \begin{bmatrix} \boldsymbol{I}_6 & \boldsymbol{0}_{6,6n} \\ \boldsymbol{I}_6 & \boldsymbol{0}_{6,6n} \\ \boldsymbol{0}_{6n,6} & \boldsymbol{I}_{6n} \end{bmatrix}^\top = \boldsymbol{\mathcal{A}}^\top \boldsymbol{P}^-(k|k) \boldsymbol{\mathcal{A}} \tag{20}$$

where $\boldsymbol{P}^-(k|k)$ is the posterior covariance before initialization and $\boldsymbol{P}(k|k)$ is the augmented covariance.

In Fig. 3, we compare the average position errors and the average orientation errors for different values of the parameters $\{\sigma_{u,t}, \sigma_{u,\theta}, \sigma_{z,t}, \sigma_{z,\theta}\}$. Notice that the error definition mirrors the one used in the theoretical development. In particular, the *average position error* is $\epsilon_t(k) \doteq \sum_{i\in\Omega} \|\tilde{\boldsymbol{t}}_i(k)\| = \sum_{i\in\Omega} \|\boldsymbol{t}_i(k) - \hat{\boldsymbol{t}}_i(k|k)\|$, whereas the *average orientation errors* is $\epsilon_\theta(k) \doteq \sum_{i\in\Omega} \|\tilde{\boldsymbol{\theta}}_i(k)\|$, where $\tilde{\boldsymbol{\theta}}_i(k)$ is the rotation vector corresponding to the error quaternion $\tilde{\boldsymbol{q}}_i(k) = \hat{\boldsymbol{q}}_i(k|k)^{-1} \boldsymbol{q}_i(k)$. It is possible to notice the effect of the first loop closing, occurring approximately at time step 1000: it leads to a drastic error reduction and the estimation error continues decreasing since the robot is traversing an already mapped area.

The consistency of our approach is evaluated by computing the *Normalized Estimation Error Squared* (NEES):[5]

$$\text{NEES}_k \doteq \tilde{\boldsymbol{x}}^\top(k|k) \boldsymbol{P}^{-1}(k|k) \tilde{\boldsymbol{x}}(k|k) \leq \chi^2_{\beta,1-\alpha}$$

where $\tilde{\boldsymbol{x}}(k|k)$ and $\boldsymbol{P}^{-1}(k|k)$ are the estimation error and the corresponding covariance matrix computed with the proposed approach, and $\chi^2_{\beta,1-\alpha}$ is the quantile of the χ^2 distribution with $\beta = \dim(\tilde{\boldsymbol{x}}(k|k))$ degrees of freedom and upper tail probability equal to α. We notice that, when a new landmark is initialized, the covariance matrix becomes rank deficient, since the *augmentation matrix* $\boldsymbol{\mathcal{A}}$, in Eq.

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

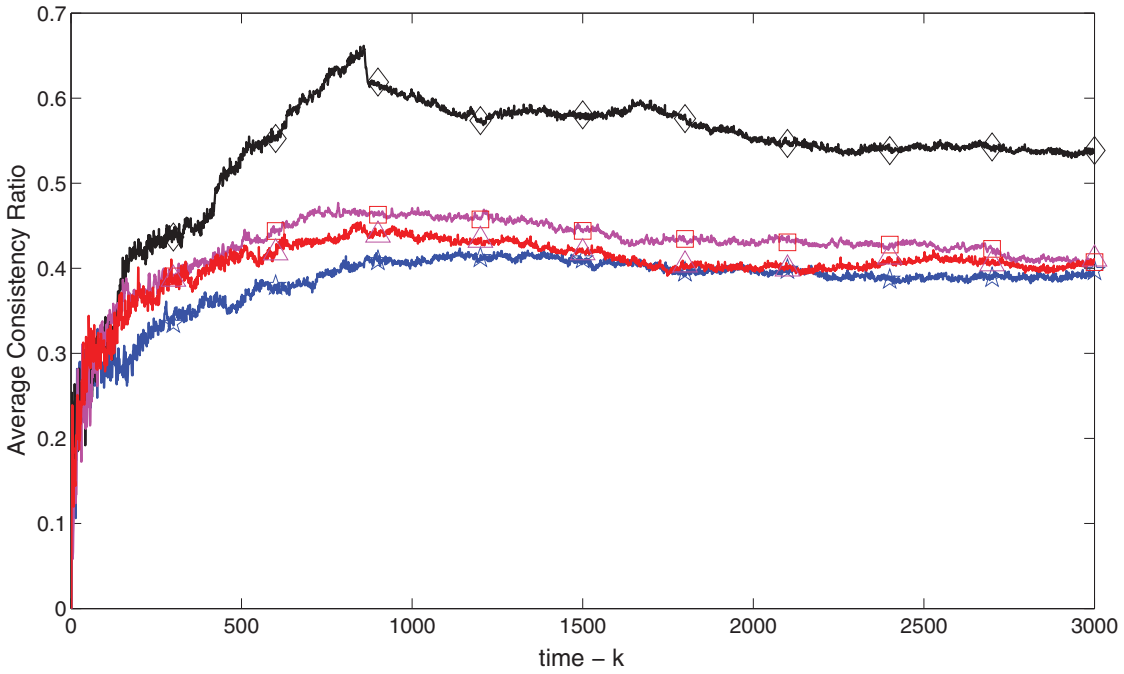

Fig. 4. Numerical results: Average consistency ratio ($\alpha = 0.01$) for different values of uncertainty. Blue (★ marker): $\{\sigma_{u,t} = 0.1\text{ m}, \sigma_{u,\theta} = 1^o, \sigma_{z,t} = 0.1\text{ m}, \sigma_{z,\theta} = 1^o\}$; Red (△ marker): $\{\sigma_{u,t} = 1\text{ m}, \sigma_{u,\theta} = 10^o, \sigma_{z,t} = 0.1\text{ m}, \sigma_{z,\theta} = 1^o\}$; Magenta (□ marker): $\{\sigma_{u,t} = 0.1\text{ m}, \sigma_{u,\theta} = 1^o, \sigma_{z,t} = 1\text{ m}, \sigma_{z,\theta} = 10^o\}$; Black (◇ marker): $\{\sigma_{u,t} = 1\text{ m}, \sigma_{u,\theta} = 10^o, \sigma_{z,t} = 1\text{ m}, \sigma_{z,\theta} = 10^o\}$.

(20), produces a linear combination of the rows in the original matrix $\boldsymbol{P}^-(k|k)$. This is a consequence of the fact that the landmark is initialized at the current pose of the robot and does not cause further complications in the algorithm. In Fig. 4 we report the *consistency ratio* $(\text{NEES}/\chi^2_{\beta,1-\alpha})^5$ averaged over 10 Monte Carlo runs for different values of $\{\sigma_{u,t}, \sigma_{u,\theta}, \sigma_{z,t}, \sigma_{z,\theta}\}$, with $\alpha = 0.01$. According to the considerations on landmark initialization, we do not report the consistency ratio in the instants corresponding to the initialization. It is possible to see that the consistency ratio remains below 1, confirming the consistency of the approach. Consistency is connected with the results on observability presented in Section 5: the presence of unobservable subspaces leads to an unbounded growth of the uncertainty in some directions of the state space, producing inconsistency in the estimation process. Accordingly, when the robot does not observe the initial pose (origin of the reference frame) the consistency ratio increases (see the initial *ramp* in Fig. 4) remarking the lack of local observability; as expected, this issue becomes more and more evident when increasing the uncertainty in the system. As soon as the first landmark is re-observed (*loop closing* occurs approximately at $k = 1000$) the consistency ratio stabilizes and remains below 1. It is clear that for longer loops or higher uncertainty the initial lack of observability would lead to inconsistency; however, we are already considering a challenging scenario, since the highest slope in the consistency ratio is obtained with $\sigma_{u,t} = \sigma_{z,t} = 1$ m and $\sigma_{u,\theta} = \sigma_{z,\theta} = 10^o$, which are large values for common applications.

6.2. Real experiments

In a real scenario, the computation of the relative pose measurements requires the following steps:

1. *Information extraction*: salient elements are extracted from sensor data (e.g., visual features from cameras, positions of distinguishable objects, corners, normal to walls, etc.);
2. *Information synthesis*: the data extracted from 3D sensors are expressed in terms of 3D vectors (e.g., 3D points, normal to planes, line directions, etc.), each one described by a unique *descriptor*;
3. *Data association*: the vectors extracted from the current observation are compared with the previous observations in order to establish the *correspondences*, i.e., the matches between each feature in the current observation and vectors occurring in previous measurements;

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

4. *Vector Registration*: the relative pose (equivalently, the roto-translation) between pairs of poses from which two observation were made is retrieved, e.g., using the registration algorithm in Appendix D.

The concept of *descriptor* is purposely kept general. If the 3D vectors correspond to visual features, *computer vision* literature offers a large range of choices (e.g., SURT, SIFT[32]). Descriptors for lines and surfaces may include both geometric and appearance-based description of the landmark (e.g., color of the plane, length of the line, area and shape of the surface).

A crucial point in the implementation of the algorithm lies in *data association*. Since the descriptors are affected by uncertainty, it is often convenient to compare each vector in the current observation with all the possible matching candidates and select the optimal match according to some criterion (e.g., *double consistency check* for visual features[54]). The set of candidate matches may be huge, since the features observed at current time need be compared with all the previous observations corresponding to landmarks in the state vector. When using vision sensors, a fast approach for detecting correspondences between current and past observations is the so called *Vocabulary Tree*, proposed by Nister in ref. [48]. A vocabulary tree is an appearance-based method for detecting similarity between two images. Suppose to have an image that contains a collection of visual features, each one denoted by a *feature descriptor*. The vocabulary tree organizes the features in a tree, according to a K-means algorithm and computes a *descriptor* of the image.[53] Therefore, the vocabulary tree associates a single identifier to each image acquired at time k, with $k = 1, 2, \ldots$, instead of having several feature descriptors for each image. Moreover, the advantage is that the *similarity score* between two images can be compared with simple operations between the corresponding image descriptors (e.g., dot product[53]).

In our setup, we use the vocabulary tree as follows. As soon as a landmark is initialized in the delayed-state architecture, the corresponding image is stored by the robot, and the image descriptor is computed. For detecting loop closing, the similarity between the current image and all the previous images is evaluated, by computing the dot product between the corresponding image descriptors. If the similarity score between the image acquired at time k and image corresponding to landmark j exceeds a given threshold, the j-th landmark is a good candidate for loop closing, i.e., it is likely that the robot is re-observing the landmark. Vocabulary tree approach is convenient in this context since it allows fast loop closing detection (the computation of the similarity scores requires few milliseconds also for large datasets[48]). Moreover, an online implementation is available.[10]

Since the loop closing detection only accounts for appearance-based information, we found it convenient to improve robustness of the detection by including geometric information from EKF: the SLAM posterior contains information of the poses of the landmarks and the relative uncertainty, hence it allows to check how likely it is that the current observation matches a given landmark. This test was implemented using a standard *validation region approach*,[5] in which measurements with high *normalized innovation squared* (NIS $\doteq \tilde{\boldsymbol{r}}(k)^\top \boldsymbol{P}_r(k)\tilde{\boldsymbol{r}}(k)$) are discarded. The overall architecture used for the real experiments is summarized in Fig. 7.

We now have all the ingredients for presenting the real experiments. We tested the proposed framework on a real dataset from the *Rawseeds* project.[17] This allows to use standard metrics for evaluating the approach in terms of estimation errors and correctness of the estimated 3D map. For the prediction phase we used the self-motion measurements from wheel odometry, while the update phase was based on the relative pose measurements from 3D vectors registration, where each vector was a visual feature (SURF) extracted from the left image of the stereo pair and then reprojected in 3D using standard stereo triangulation.

The trajectory estimated with the proposed approach is shown as a blue line in Fig. 5. The corresponding ground truth trajectory (obtained using a *RTK*-GPS[27]) is shown in red, in the same figure. Figure 6 reports the candidate loop closings detected by the vocabulary tree and loop closings that passed the NIS test and that were actually included in the EKF-SLAM update phase. A quantitative evaluation of the approach can be obtained from the statistics in Fig. 8. Such metrics are computed with the *metrics computation toolkit* available at ref. [17]. In Figure 8 they are reported the Cartesian errors (considering the x and the y components of the error separately) and the orientation error (yaw angle θ only) since the dataset corresponds to a planar problem. We obtained the planar estimates by projecting the 3D estimates on the ground plane. We remark that the peaks in the orientation statistics are only due to the fact that the ground truth trajectory has more poses than the estimated trajectory (as

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

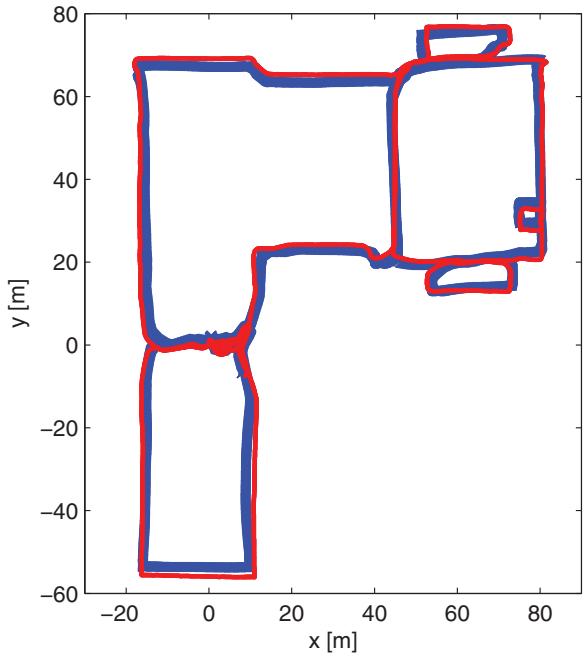

Fig. 5. Real test: the ground truth trajectory from *Rawseeds* (dataset *Bovisa 20090225b*) is shown as a red line, whereas the trajectory estimated with the proposed approach is plotted in blue.

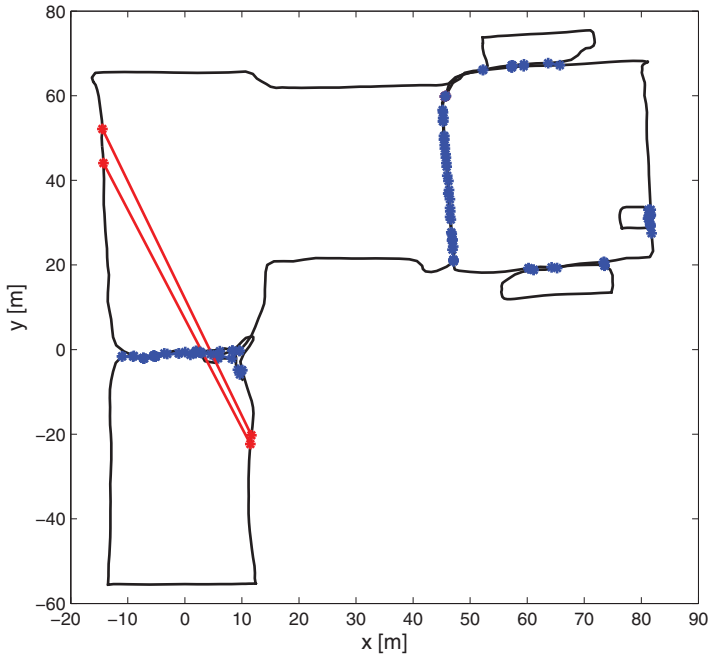

Fig. 6. Estimated robot trajectory (black line). The blue and the red segments denote the candidate loop closings: the segments connect the poses at which the images matched by the vocabulary tree were acquired. The blue segments are the loop closings that passed the NIS test and were included in filter update, whereas the red segments were classified as outliers and discarded.

mentioned before the filter initializes a new landmark every 20 time steps) and the missing samples in the estimated trajectory are estimated by interpolating contiguous samples. We refer to Fig. 9 for further discussion.

In Fig. 10(a), instead, it is shown a statistic on the Cartesian error over the whole experiment: the plot reports (i) the mean error, (ii) the histogram with the frequency with which a certain error occurs,

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

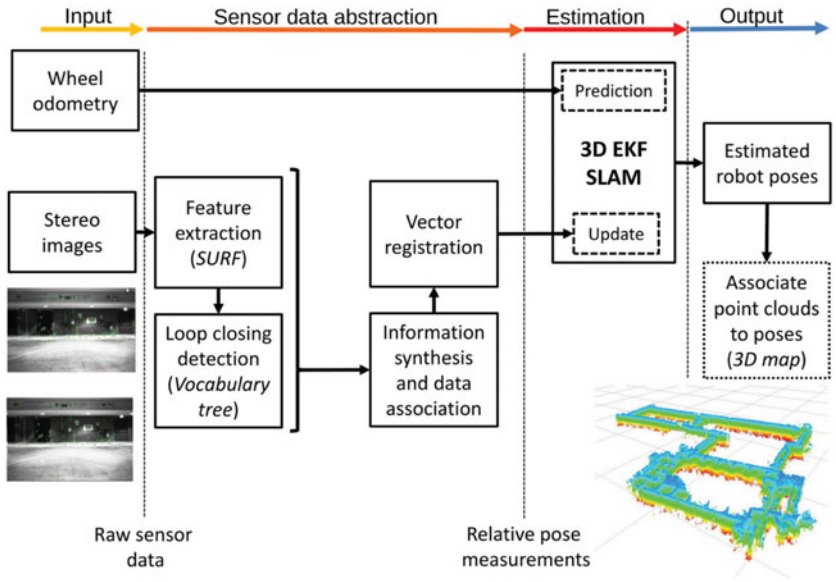

Fig. 7. Block diagram describing the architecture used in the real experiments. The scheme shows how raw sensors data are abstracted in terms of relative pose measurements and then fed to the proposed EKF-SLAM approach. In particular, the computation of relative poses from a sequence of stereo images starts from the extraction of features from the images (SURF in our experiments). The features are also used to detect similarity with previous images, using a vocabulary tree (loop closing detection). Once a loop closing with a landmark j is detected, the features from the corresponding stereo images are converted into two sets of 3D vectors (one corresponding to the current observation, the other one describing the observations taken from pose j) and a vector registration algorithm (Appendix D) is used to compute the relative transformation between the current pose and the pose of landmark j.

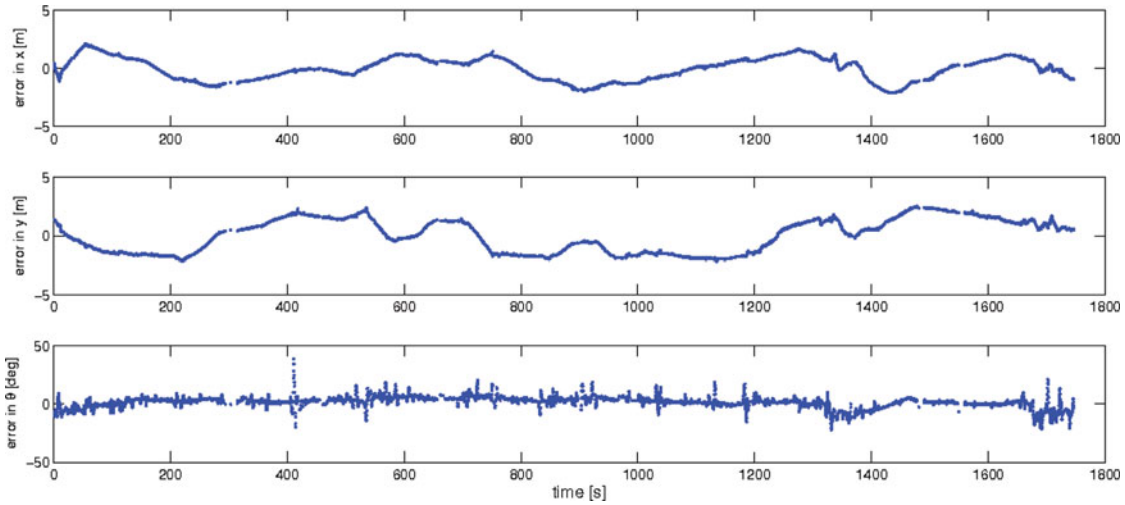

Fig. 8. Real test: Cartesian error along the x-axis and the y-axis; orientation error in the yaw angle θ. The errors are reported as function of time, expressed in seconds.

(iii) the 3σ intervals that describe the region in which the error is likely to fall. We also report in Fig. 10(b) the corresponding error statistics of a benchmark solution of Rawseeds, that is based on standard EKF-SLAM and stereo data (stereo-based CI-graph SLAM[50]).

A further advantage of the considered framework is that pose estimation allows to retrieve a dense representation of the map. For instance we can obtain a dense point cloud representation from a stereo pair acquired at a given time step and we can use the corresponding pose estimate for roto-translating the dense point cloud, hence expressing all the points in a common reference frame. The outcome of the mapping process is shown in Fig. 11 using the Rawseeds dataset.

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

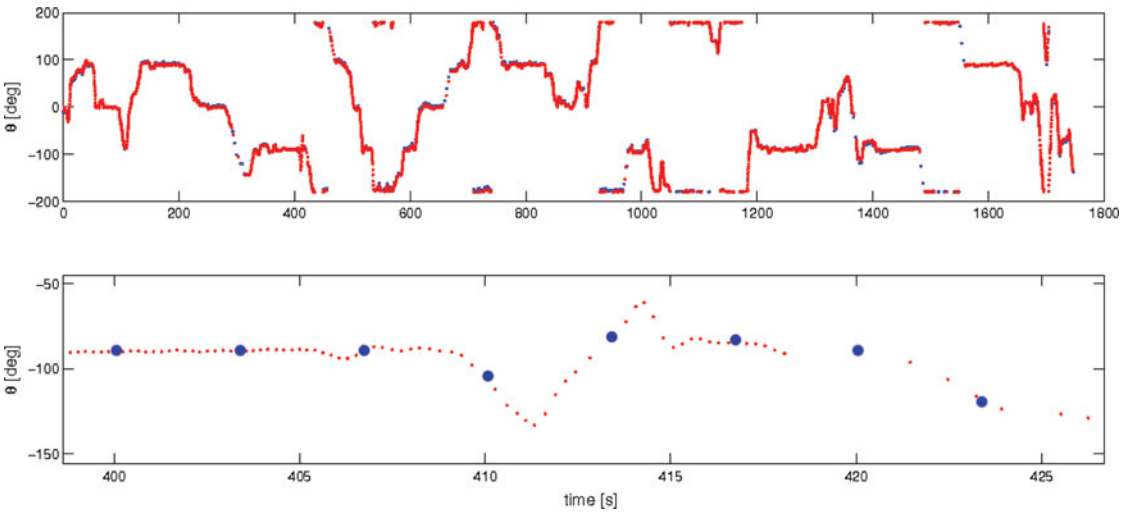

Fig. 9. Real test: yaw angle θ of the ground truth (red samples) versus estimated yaw (blue samples). The upper plot does not reveal remarkable differences between the two angles, whereas the zoom in the lower plot allows to understand the nature of the error peaks that can be observed in Fig. 8: the estimated orientation (blue dots) has fewer samples than the ground truth (in red) and when a ground truth angle does not have a corresponding estimate, an estimate is computed by linear interpolation between the contiguous estimates (nearest blue dots). It may be now understood that the spikes in the orientation error are only due to the benchmarking technique and are not related to the accuracy of the proposed approach.

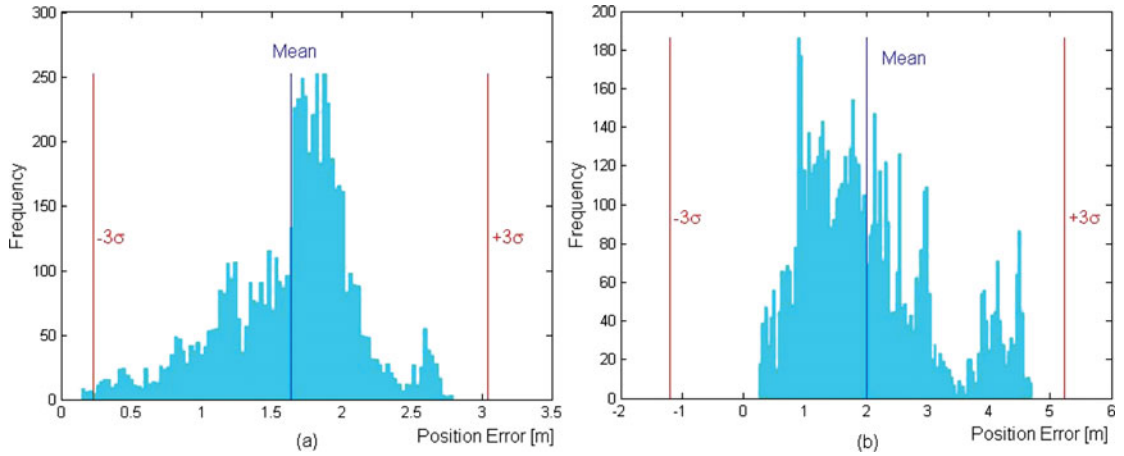

Fig. 10. Real test: histogram plots showing the frequency of Cartesian errors for the proposed approach (a) versus the CI-graph SLAM approach (b). The figure also reports the mean errors and the 3σ confidence intervals for the Cartesian errors.

6.3. Relative pose measurements and 3D objects

This last experimental section is aimed at showing that the considered framework is not only functional to a delayed-state architecture but has interesting applications in a more traditional EKF-SLAM setup in which the landmarks are external objects in the environment. In this EKF formulation, we store in the state vector the poses of 3D bodies corresponding to distinguishable objects, instead of storing only their positions, as it happens in standard EKF-SLAM implementations.[2,42] The advantage in this case is the fully observability of the approach, with respect to the standard EKF-SLAM observability results:[34] the approach is expected to be more stable and the consistency of the estimation process assures a more reliable loop closing detection. Moreover, we focus on a full 3D case to further test the robustness of the quaternion representation.

Let us consider the case of a UAV (*Unmanned Aerial Vehicle*) starting an a given pose and performing SLAM to support landing.[46] In our toy problem the vehicle approaches the ground rotating around its x axis (red arrow in Fig. 12) and describing a spiral. The ground truth landing

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

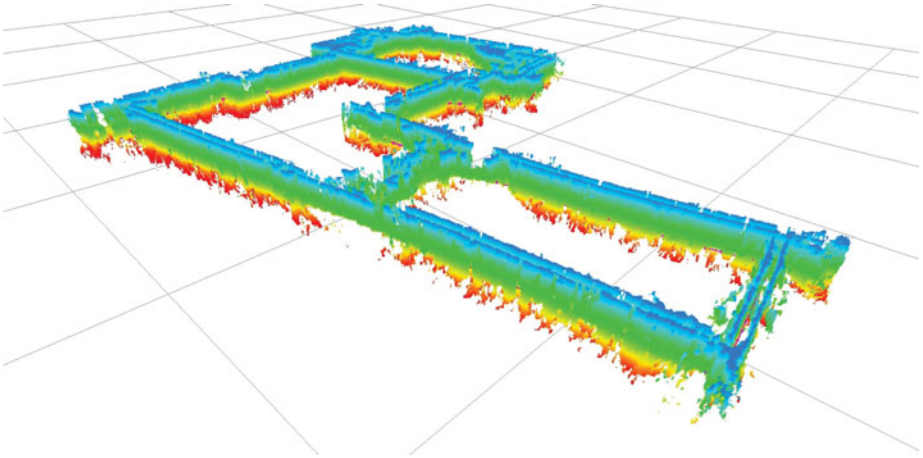

Fig. 11. Real test: dense 3D map of the scenario computed from the Rawseeds dataset. A stereo pair is associated to each pose in the delayed-state filter and a dense point cloud is computed using standard stereo-vision techniques.[32] Then each point cloud is roto-translated according to the pose estimates obtained from the delayed-state EKF-SLAM, and the result is a consistent dense representation of the environment.

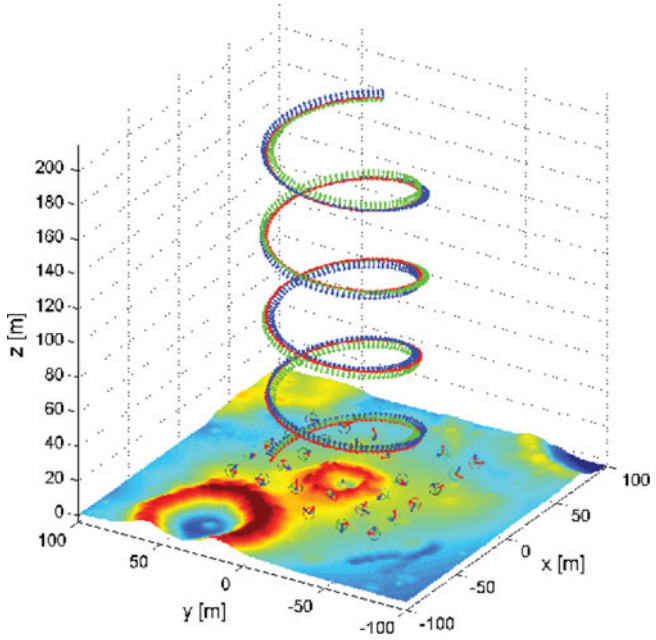

Fig. 12. Numerical results: ground truth trajectory of an Unmanned Aerial Vehicle performing SLAM during landing phase. The trajectory is shown as a sequence of poses, each one associated with a reference frame, whereas distinguishable objects on the ground are denoted with circles and a reference frame is associated to each of them.

trajectory is shown in Fig. 12 as a sequence of poses, each denoted by a reference frame. The same figure shows that on the ground are deployed some distinguishable objects (the objects are labeled with circles and a reference frame is attached to each of them). The robot is able to discern such objects and can take noisy measurements of their poses. Moreover it can take noisy odometric measurements, using, for instance, visual odometry or an inertial navigation system.

It is easy to see that the approach proposed in this paper can be adapted to this problem with few changes. The only step that needs slight modifications is the landmark initialization, since now a landmark is not initialized at the current robot pose but at the pose of the observed object. Moreover, it is easy to see that the observability analysis remains unchanged if the first observed landmark is set to be the origin of the reference frame. In Fig. 13 we report the trajectory estimated with the proposed approach, compared with the ground truth. For a quantitative evaluation, in Fig. 14 we report the

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

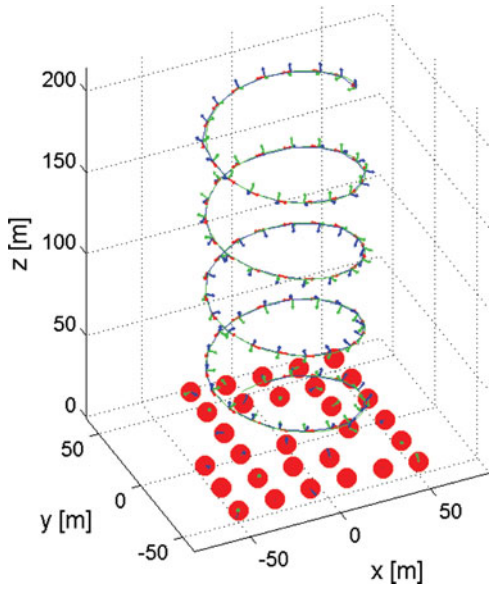

Fig. 13. Numerical results: ground truth trajectory (green) versus trajectory estimated with the proposed approach (blue). The figure also reports the estimated poses of the objects in the map.

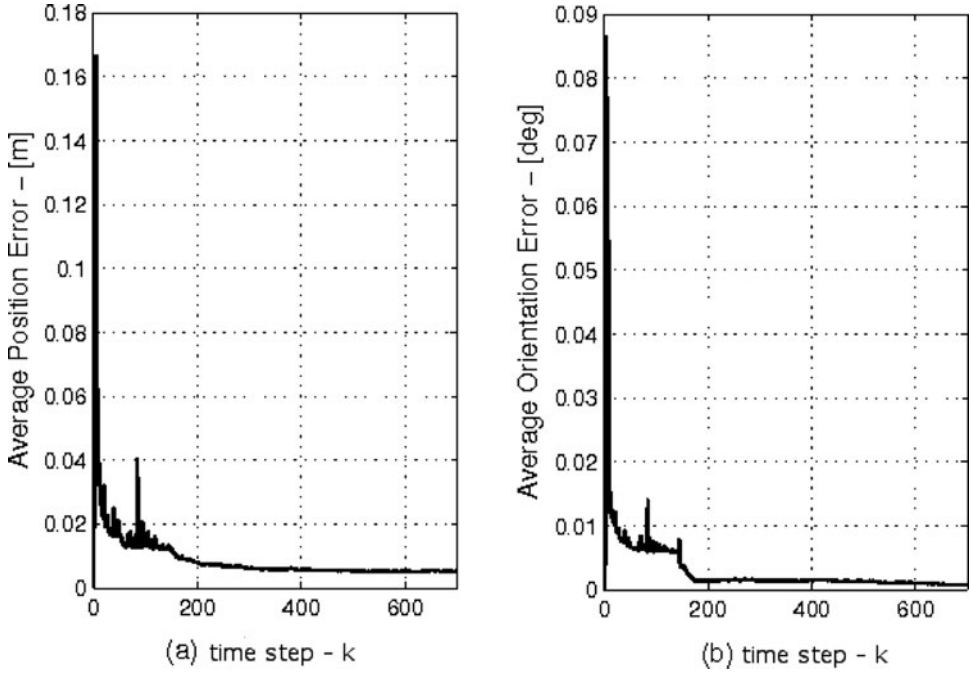

Fig. 14. Numerical results: Average position and orientation errors for $\{\sigma_{u,t} = 0.5$ m, $\sigma_{u,\theta} = 0.5^o$, $\sigma_{z,t} = 1$ m, $\sigma_{z,\theta} = 1^o\}$.

average position and the average orientation errors (that take into account both the error on robot pose and on landmark poses).

We conclude by observing that is possible to envision several other applications of the pose-based SLAM discussed in this section; for instance, the 3D bodies in the environment can be cars in a parking lot or building corners when the robot is moving in urban environment; in indoor environment an effective application can use the door frames as steady landmarks, for its geometric structure that is easily recognizable in camera images. Moreover, such applications may rely on pose estimation algorithms that have been extensively studied in the context of computer vision[21] and 3D grasping.[3] We strongly believe that this constitutes a remarkable advantage of the proposed

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

technique: the observability properties and the quaternion-based formulation enhances the robustness of the approach, whereas the use of 3D bodies as landmarks provides more information to solve in a robust fashion data association and landmark detection problems.

7. Conclusion

We can identify three main contributions in this work. The first regards the derivation of a quaternion-based approach for solving Simultaneous Localization and Mapping from relative pose measurements. The approach does not use heuristics for forcing the unit norm constraint on quaternions of rotation and it is shown to be a high-level abstraction of several 3D perceptions in mobile robotics, being suitable for information fusion from heterogeneous sensors. Furthermore it has a direct application within a delayed-state architecture for SLAM and its extension to the multi robot case is straightforward.

The second contribution is the observability analysis of the quaternion-based filter. The work formally proves that the SLAM system is completely observable over a given interval of time, as long as this interval includes an observation of the *anchor*, i.e., the landmark that was set at the origin of the reference frame for mapping.

The third contribution regards the experimental analysis of the approach in different application scenarios. The analysis includes a numerical evaluation for assessing the consistency of the SLAM approach and tests on a real dataset that allow to evaluate the accuracy of the approach using standard benchmarking metrics.

Acknowledgments

This work was partially funded by Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) under MEMONET National Research Project.

References

1. F. Aghili, "3D simultaneous localization and mapping using IMU and its observability analysis," *Robotica* **29**(6), 805–814 (2010).
2. J. Artieda, J. M. Sebastian, P. Campoy, J. F. Correa, I. F. Mondragon, C. Martinez and M. Olivares, "Visual 3-d SLAM from UAVs," *J. Intell. Robot. Syst.* **55**(4-5), 299–321 (2009).
3. P. Azad, T. Asfour and R. Dillmann, "Stereo-Based 6D Object Localization for Grasping with Humanoid Robot Systems," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2007) pp. 919–924.
4. T. Bailey, J. Nieto, J. Guivant, M. Stevens and E. Nebot, "Consistency of the EKF-SLAM Algorithm," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2006) pp. 3562–3568.
5. Y. Bar-Shalom, X. R. Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation* (John Wiley and Sons, 2001).
6. E. Bayro-Corrochano and L. E. Falcon-Morales, "Geometric algebra of points, lines, planes and spheres for computer vision and robotics," *Robotica* **23**(6), 755–770 (2005).
7. E. Bayro-Corrochano, L. E. Falcon-Morales and J. Zamora-Esquivel, "Visually Guided Robotics Using Conformal Geometric Computing," **In**: *Mobile Robots: Perception & Navigation*, (InTech, 2007).
8. W. G. Breckenridge, "Quaternions - proposed standard conventions," In" *JPL, Tech. Rep. INTEROFFICE MEMORANDUM IOM 343-79-1199* (1999).
9. M. Bryson and S. Sukkarieh, "Observability analysis and active control for airborne SLAM". *IEEE Transactions on Aerospace and Electronic System* **44**(1), 261–280 (2008).
10. C. Cadena, D. Galvez-Lopez, F. Ramos, J. D. Tardós and J. Neira, "Robust Place Recognition With Stereo Cameras," *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010) pp. 5182–5189.
11. G. Calafiore and B. Bona, "Constrained optimal fitting of three-dimensional vector patterns," *IEEE Trans. Robot. Autom.* **14**(5), 838–844 (1998).
12. L. Carlone and B. Bona, "On registration of Uncertain Three-Dimensional Vectors with Application to Robotics," *Proceedings of the IFAC World Congress* (2011).
13. L. Carlone, V. Macchia, F. Tibaldi and B. Bona, "Robot Localization and 3D Mapping: Observability Analysis and Applications," *Proceedings of the Int. Symposium on Artificial Intelligence, Robotics and Automation in Space* (2012).
14. H. Casarrubias-Vargas, A. H. Petrilli-Barceló and E. Bayro-Corrochano, "EKF-SLAM and Machine Learning Techniques for Visual Robot Navigation," *Proceedings of the 20th International Conference on Pattern Recognition* (2010) pp. 396–399.

15. J. A. Castellanos, R. Martinez-Cantin, J. D. Tardós and J. Neira, "Robocentric map joining: Improving the consistency of EKF-SLAM," *Robot. Auton. Syst.* **55**(1), 21–29 (2007).
16. J. A. Castellanos, J. Neira and J. D. Tardós, "Limits to the Consistency of EKF-Based SLAM," *5th IFAC Symp. on Intelligent Autonomous Vehicles* (2004) pp. 1244–1249.
17. S. Ceriani, G. Fontana, A. Giusti, D. Marzorati, M. Matteucci, D. Migliore, D. Rizzi, D. G. Sorrenti and P. Taddei, "RAWSEEDS ground truth collection systems for indoor self-localization and mapping," *Auton. Robots J.* **27**(34), 353–371 (2009).
18. H. Chiu, S. Williams, F. Dellaert, S. Samarasekera and R. Kumar, "Robust Vision-Aided Navigation Using Sliding-Window Factor Graphs," *Proceedings of the IEEE International Conference on Robotics and Automation* (2013).
19. D. M. Cole and P. M. Newman, "Using Laser Range Data for 3D SLAM in Outdoor Environments," *IEEE International Conference on Robotics and Automation* (2006) pp. 1556–1563.
20. A. Cristofaro, A. Renzaglia and A. Martinelli, "Distributed Information Filters for MAV Cooperative Localization," *Proceedings of the 10th International Symposium on Distributed Autonomous Robotics Systems* (2010) pp. 287–293.
21. D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *Int. J. Comput. Vis.* **15**(1-2), 123–141 (1995).
22. A. Doucet, N. de Freitas, K. Murphy and S. Russel, "Rao-Blackwellized Particle Filtering for Dynamic Bayesian Networks," *Proceedings of the Conf. on Uncertainty in Artificial Intelligence* (2000) pp. 176–183.
23. H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I," *Robot. Autom. Mag.* **13**, 99–110 (2006).
24. H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part II," *Robot. Autom. Mag.* **13**, 108–117 (2006).
25. H. F. Durrant-Whyte, "Uncertain geometry in robotics," *IEEE Trans. Robot. Automat.* **4**(1), 22–31 (1988).
26. C. Estrada, J. Neira and J. D. Tardós, "Hierarchical SLAM: Real-time accurate mapping of large environments," *IEEE Trans. Robot.* **21**(4), 588–596 (2005).
27. D. Sorrenti, W. Burgard, G. Grisetti, M. Ruhnke, C. Stachniss, G. Fontana, M. Matteucci, D. Marzorati, J. D. Tardós, *et al.* "Rawseed Project: Deliverable D4.1, Benchmark problems", Technical Report, available online at: http://www.rawseeds.org/home/category/documents/deliverables/, 2009.
28. R. M. Eustice, H. Singh and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *Int. J. Robot. Res.* **22**(6), 1100–1114 (2006).
29. M. A. Garcia and A. Solanas, "3D Simultaneous Localization and Modeling from Stereo Vision," *IEEE International Conference on Robotics and Automation* (2004) pp. 847–853.
30. D. Goshen-Meskin and I. Y. Bar-Itzhack, "Observability analysis of piece-wise constant systems, Part 1: Theory," *IEEE Trans. Aerosp. Electron. Syst.* **28**(4), 1056–1067 (1992).
31. D. Goshen-Meskin and I. Y. Bar-Itzhack, "Observability analysis of piece-wise constant systems, Part 2: Application to inertial navigation in-flight alignment," *IEEE Trans. Aerosp. Electron. Syst.* **28**(4), 1068–1075 (1992).
32. R. I. Hartley and A. Zisserman," *Multiple View Geometry in Computer Vision* (Cambridge University Press, 2000) ISBN: 0521623049.
33. R. A. Horn and C. R. Johnson, *Matrix Analysis* (Cambridge University Press, UK, 1985).
34. G. Huang, A. I. Mourikis and S. I. Roumeliotis, "Observability-based rules for designing consistent EKF-SLAM estimators," *Int. J. Robot. Res.* **29**(5), 502–528 (2010).
35. S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended kalman filter based SLAM," *IEEE Trans. Robot.* **23**(5), 1036–1049 (2001).
36. V. Ila, J. M. Porta and J. Andrade-Cetto, "Information-based compact pose SLAM," *IEEE Trans. Robot.* **26**(1), 78–93 (2010).
37. S. Julier and J. Uhlmann, "A Counter Example to the Theory of Simultaneous Localization and Map Building," *Proceedings of the IEEE International Conference on Robotics and Automation* (2001) pp. 4238–4243 .
38. M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard and F. Dellaert, "iSAM2:incremental smoothing and mapping using the bayes tree," *Int. J. Robot. Res.* **31**, 217–236 (2012).
39. J. H. Kim and S. Sukkarieh, "Improving the real-time efficiency of inertial SLAM and understanding its observability," *IEEE International Conference on Intelligent RObots and Systems*, pp. 1264–1269 (2004).
40. J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality* (Princeton University Press, 1999).
41. E. J. Lefferts, F. L. Markley and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *J. Guid. Control Dyn.* **5**(5), 417–429 (1982).
42. T. Lemaire, C. Berger, I. K. Jung and S. Lacroix, "Vision-based SLAM: Stereo and monocular approaches," *Int. J. Comput. Vis.* **74**(3), 343–364 (2007).
43. J. Leonard, R. Rikoski, P. Newman and M. Bosse, "Mapping partially observable features from multiple uncertain vantage points," *Int. J. Robot. Res.* **21**(10), 943–975 (2002).
44. F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots* **4**, 333–349 (1997).
45. J. L. Marins, X. Yun, E. R. Bachman and R. B. McGhee, "An Extended Kalman Filter for Quaternion-Based Orientation Estimation using MARG Sensors," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (2001) pp. 2003–2011 .

46. A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. Johnson, A. Ansar and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Trans. Robot.* **25**(2), 264–280 (2009).
47. P. Newman, On the Structures and Solution of Simultaneous Localization and Mapping Problem, *Ph.D. thesis* (Department of Mathematical Sciences, Division of Mathematics, Chalmers University of Technology and Göteborg University, 2007).
48. D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2005) pp. 2161–2168.
49. A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg and S. Thrun, "6D SLAM with an Application in Autonomous Mine Mapping," *IEEE International Conference on Robotics and Automation* (2004) pp. 1998–2003.
50. P. Piniés, L. M. Paz and J. D. Tardós, "CI-graph SLAM for 3D reconstruction of large and complex environments using a multicamera system," *Int. J. Field Robot.* **27**(5), 561–586 (2010).
51. P. Piniés and J. D. Tardós, "Scalable SLAM Building Conditionally Independent Local Maps," *IEEE-RSJ International Conference on Intelligent Robots and Systems* (2007).
52. A. M. Sabatini, "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Trans. Biomed. Eng.* **53**(7), 1346–1356 (2006).
53. D. Sabatta, D. Scaramuzza and R. Siegwart, "Improved Appearance-Based Matching in Similar and Dynamic Environments using a Vocabulary Tree," *Proceedings of the IEEE International Conference on Robotics and Automation* (2010) pp. 2262–2269 .
54. D. Scaramuzza and R. Siegwart, "Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Trans. Robot.* **24**(5) (2008).
55. M. D. Shuster, "A survey of attitude representations," *J. Astronaut. Sci.* **41**(4), 439–517 (1993).
56. B. Siciliano and O. Khatib," *Handbook of Robotics* (Springer, 2008).
57. R. Smith and P. Cheesman, "On the representation of spatial uncertainty," *Int. J. Robot. Res.* **5**(4), 56–68 (1987).
58. C. Stachniss, G. Grisetti, D. Hahnel and W. Burgard, "Improved Rao-Blackwellized Mapping by Adaptive Sampling and Active Loop-Closure," *Proceedings of the Workshop on Self-Organization of AdaptiVE behavior* (2004) pp. 1–15.
59. S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte and A. Y. Ng, "Simultaneous Mapping and Localization with Sparse Extended Information Filters," *Proceedings of the 5th Int. Workshop on Algorithmic Foundations of Robotics* (2002).
60. N. Trawny and S. I. Roumeliotis, Indirect Kalman Filter for 3D Attitude Estimation, *Tech. Rep.* 2005-002 (University of Minnesota, Dept. of Comp. Sci. and Eng. 2005).
61. M. Walter, R. Eustice and J. Leonard, "Exactly sparse extended information filters for feature-based SLAM," *Int. J. Robot. Res.* **26**(4), 335–359 (2007).
62. J. Weingarten and R. Siegwart, "EKF-Based 3D SLAM for Structured Environment Reconstruction," *Proceedings of the International Conference on Intelligent Robots and Systems* (2005) pp. 2–6.

Appendix A: Preliminaries on Quaternions and Rotations

This section recalls basic concepts on unit quaternions, as described in ref. [40] (apart from Eq. (A3) that can be easily verified by inspection). Depending on the context a quaternion will be denoted as a column vector $\boldsymbol{q} = [q_1\ q_2\ q_3\ q_4]^\top \in \mathbb{R}^4$ or as an ipercomplex number, i.e., $\boldsymbol{q} = \boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3 + q_4$, with $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ satisfying:

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$$
$$\boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k} \qquad \boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i} \qquad \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j}$$

Unit quaternions, i.e., quaternions having unit norm, are compact representations for rotations and have been used in several applications concerning 3D navigation, see.[8,41,55,60] A quaternion can be also written as $\boldsymbol{q} = [\boldsymbol{v}^\top\ s]^\top$, where the first element $\boldsymbol{v} \in \mathbb{R}^3$ is the *vector part* of the quaternion whereas the last element is the *real part* (contrarily to related work we postpone the real part to the vector part so to have a unified treatment of quaternions and Cartesian vectors in homogeneous coordinates). Furthermore, unit quaternions can be written as $\boldsymbol{q} = [\sin(\vartheta/2)\boldsymbol{u}^\top\ \cos(\vartheta/2)]^\top$, and describe a rotation of an angle ϑ around the axis $\boldsymbol{u} \in \mathbb{R}^3$, with $\|\boldsymbol{u}\| = 1$. Then we can recall the following facts.

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

Quaternions multiplication. Given two quaternions $\boldsymbol{q}^a = [q_1^a\ q_2^a\ q_3^a\ q_4^a]^\top$ and $\boldsymbol{q}^b = [q_1^b\ q_2^b\ q_3^b\ q_4^b]^\top$, the quaternion product $\boldsymbol{q}^c = \boldsymbol{q}^a\boldsymbol{q}^b$ can be computed as:

$$\boldsymbol{q}^c = \begin{bmatrix} q_4^a & -q_3^a & q_2^a & q_1^a \\ q_3^a & q_4^a & -q_1^a & q_2^a \\ -q_2^a & q_1^a & q_4^a & q_3^a \\ -q_1^a & -q_2^a & -q_3^a & q_4^a \end{bmatrix} \begin{bmatrix} q_1^b \\ q_2^b \\ q_3^b \\ q_4^b \end{bmatrix} = \boldsymbol{\Upsilon}\boldsymbol{q}^b$$

or, equivalently, as:

$$\boldsymbol{q}^c = \begin{bmatrix} q_4^b & q_3^b & -q_2^b & q_1^b \\ -q_3^b & q_4^b & q_1^b & q_2^b \\ q_2^b & -q_1^b & q_4^b & q_3^b \\ -q_1^b & -q_2^b & -q_3^b & q_4^b \end{bmatrix} \begin{bmatrix} q_1^a \\ q_2^a \\ q_3^a \\ q_4^a \end{bmatrix} = \bar{\boldsymbol{\Upsilon}}\boldsymbol{q}^a$$

Inverse quaternion. Given a unit quaternion in vector-scalar form $\boldsymbol{q} = [\boldsymbol{v}^\top s]^\top$, the inverse quaternions is given by $\boldsymbol{q}^{-1} = [-\boldsymbol{v}^\top s]^\top$. Moreover, when inverting the product of quaternions, the following rule applies:

$$(\boldsymbol{q}_a\boldsymbol{q}_b \dots \boldsymbol{q}_N)^{-1} = \boldsymbol{q}_N^{-1} \dots \boldsymbol{q}_b^{-1}\boldsymbol{q}_a^{-1}$$

Quaternion and rotation matrix. Given a unit quaternion $\boldsymbol{q} = [q_1\ q_2\ q_3\ q_4]^\top$, the corresponding rotation matrix $\boldsymbol{R}(\boldsymbol{q})$ can be computed as:

$$\boldsymbol{R}(\boldsymbol{q}) = \begin{bmatrix} q_1^2-q_2^2-q_3^2+q_4^2 & 2(q_1q_2 - q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & -q_1^2+q_2^2-q_3^2+q_4^2 & 2(q_2q_3 - q_1q_4) \\ 2(q_1q_3 - q_2q_4) & 2(q_2q_3 + q_1q_4) & -q_1^2-q_2^2+q_3^2+q_4^2 \end{bmatrix}$$

Moreover if $\boldsymbol{q}_c = \boldsymbol{q}_a\boldsymbol{q}_b$, then $\boldsymbol{R}(\boldsymbol{q}_c) = \boldsymbol{R}(\boldsymbol{q}_a)\boldsymbol{R}(\boldsymbol{q}_b)$, i.e., the quaternion product realizes the composition of the rotations corresponding to each quaternion; furthermore, given the inverse quaternion $\boldsymbol{q}^{-1}$, then $\boldsymbol{R}(\boldsymbol{q}^{-1}) = \boldsymbol{R}^\top(\boldsymbol{q})$.

Rotations and 3D vectors. Given a unit quaternion $\boldsymbol{q}$ and a vector $\boldsymbol{t} \in \mathbb{R}^3$, if we express the vector $\boldsymbol{t}$ in homogeneous coordinates as $\boldsymbol{p} = [\boldsymbol{t}^\top\ 1]^\top$, then the following property holds:

$$\boldsymbol{p}_r \doteq \boldsymbol{q}\boldsymbol{p}\boldsymbol{q}^{-1} = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{q})\boldsymbol{t} \\ 1 \end{bmatrix} \tag{A1}$$

Homogeneous sum. Given two homogeneous vectors $\boldsymbol{p}_a = [\boldsymbol{t}_a^\top\ 1]^\top$ and $\boldsymbol{p}_b = [\boldsymbol{t}_b^\top\ 1]^\top$, their sum and difference is:

$$\boldsymbol{p}_a \oplus \boldsymbol{p}_b = \begin{bmatrix} \boldsymbol{t}_a + \boldsymbol{t}_b \\ 1 \end{bmatrix} \quad \boldsymbol{p}_a \ominus \boldsymbol{p}_b = \begin{bmatrix} \boldsymbol{t}_a - \boldsymbol{t}_b \\ 1 \end{bmatrix}$$

Exponential map. The *exponential map* $\Phi : \mathbb{R}^3 \ni \boldsymbol{\theta} \to \boldsymbol{q} \in \mathrm{SO}(3)$ from the vector space $\mathbb{R}^3$ to an element of the special orthogonal group SO(3), maps the rotation vector $\boldsymbol{\theta}$ into a unit quaternion describing a rotation of an angle $\|\boldsymbol{\theta}\|$ around the axis $\boldsymbol{\theta}/\|\boldsymbol{\theta}\|$[1]. The exponential map can be written explicitly as $\Phi(\boldsymbol{\theta}) = \left[\sin\left(\frac{\|\boldsymbol{\theta}\|}{2}\right)\frac{\boldsymbol{\theta}^\top}{\|\boldsymbol{\theta}\|}\ \ \cos\left(\frac{\|\boldsymbol{\theta}\|}{2}\right)\right]^\top$.

[1] More formally, unit quaternions belong to the group SU(2), which is a double cover of SO(3), since the quaternions $\boldsymbol{q}$ and $-\boldsymbol{q}$ represent the same rotation.

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

Small-angle approximations. Given a unit quaternion $\tilde{\boldsymbol{q}}$, the following small-angle approximation holds:[60]

$$\tilde{\boldsymbol{q}} \approx \begin{bmatrix} \tilde{\boldsymbol{\theta}}/2 \\ 1 \end{bmatrix} \tag{A2}$$

being $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^3$ a vector describing a rotation of an angle $\|\tilde{\boldsymbol{\theta}}\|$ around the axis $\tilde{\boldsymbol{\theta}}/\|\tilde{\boldsymbol{\theta}}\|$.

Moreover, for small rotations $\tilde{\boldsymbol{q}}^a \approx [(\tilde{\boldsymbol{\theta}}^a)^\top \; 1]^\top$ and $\tilde{\boldsymbol{q}}^b \approx [(\tilde{\boldsymbol{\theta}}^b)^\top \; 1]^\top$ it holds the following first-order approximation:

$$\tilde{\boldsymbol{q}}^c = \tilde{\boldsymbol{q}}^a \tilde{\boldsymbol{q}}^b \approx \tilde{\boldsymbol{q}}^a \oplus \tilde{\boldsymbol{q}}^b = \begin{bmatrix} \frac{\tilde{\boldsymbol{\theta}}^a}{2} + \frac{\tilde{\boldsymbol{\theta}}^b}{2} \\ 1 \end{bmatrix} \tag{A3}$$

with $\oplus$ denoting the sum in homogeneous coordinates. Finally, for a small quaternion $\tilde{\boldsymbol{q}} \approx [\tilde{\boldsymbol{\theta}}^\top \; 1]^\top$ and the corresponding rotation matrix $\boldsymbol{R}(\tilde{\boldsymbol{q}})$, it holds:

$$\boldsymbol{R}(\tilde{\boldsymbol{q}}) = \boldsymbol{R}(\tilde{\boldsymbol{\theta}}) \approx \boldsymbol{I}_3 + \boldsymbol{S}(\tilde{\boldsymbol{\theta}}) \tag{A4}$$

being $\boldsymbol{S}([\tilde{\theta}_1 \; \tilde{\theta}_2 \; \tilde{\theta}_3]^\top)$ the skew-symmetric matrix:

$$\boldsymbol{S}(\tilde{\boldsymbol{\theta}}) \doteq \begin{bmatrix} 0 & -\tilde{\theta}_3 & \tilde{\theta}_2 \\ \tilde{\theta}_3 & 0 & -\tilde{\theta}_1 \\ -\tilde{\theta}_2 & \tilde{\theta}_1 & 0 \end{bmatrix}. \tag{A5}$$

Appendix B: Error State Motion Model

In this section we devise the expression of the motion model of the robot in terms of estimation and odometric measurement errors. Let us consider robot prediction for the orientation variables in (1) and the corresponding estimate in (2). By pairwise multiplication we obtain:

$$\hat{\boldsymbol{q}}_0^{-1}(k|k-1)\boldsymbol{q}_0(k) = (\hat{\boldsymbol{q}}_0(k-1|k-1)\bar{\boldsymbol{q}}_u(k))^{-1}\,(\boldsymbol{q}_0(k-1)\boldsymbol{q}_u(k)) \tag{B1}$$

The first member of (B1) is, by definition, the estimation error of robot orientation at time k, before including information from exteroceptive sensors, i.e., $\tilde{\boldsymbol{q}}_0(k|k-1)$. We can then develop the right hand side of (B1) as follows:

$$\tilde{\boldsymbol{q}}_0(k|k-1) = \bar{\boldsymbol{q}}_u(k)^{-1}\hat{\boldsymbol{q}}_0(k-1|k-1)^{-1}\boldsymbol{q}_0(k-1)\boldsymbol{q}_u(k)$$

Now let us substitute $\tilde{\boldsymbol{q}}_0(k-1|k-1) = \hat{\boldsymbol{q}}_0(k-1|k-1)^{-1}\boldsymbol{q}_0(k-1)$ and rewrite $\boldsymbol{q}_u(k)$ as function of the measured rotation and the corresponding error:

$$\tilde{\boldsymbol{q}}_0(k|k-1) = \left(\bar{\boldsymbol{q}}_u(k)^{-1}\tilde{\boldsymbol{q}}_0(k-1|k-1)\bar{\boldsymbol{q}}_u(k)\right)\tilde{\boldsymbol{q}}_u(k) \tag{B2}$$

The previous expression provides the relationship between the error of (prior) robot orientation estimate at time k in function of the error of robot orientation posterior at time $k-1$ and the odometric measurement error. Under the hypothesis of small angular errors we can rewrite the quaternions as

$$\tilde{\boldsymbol{q}}_0(k|k-1) \approx \left[\frac{1}{2}\tilde{\boldsymbol{\theta}}_0(k|k-1)^\top \; 1\right]^\top$$

$$\tilde{\boldsymbol{q}}_0(k-1|k-1) \approx \left[\frac{1}{2}\tilde{\boldsymbol{\theta}}_0(k-1|k-1)^\top \; 1\right]^\top$$

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

and

$$\tilde{\boldsymbol{q}}_u(k) \approx \left[\frac{1}{2}\tilde{\boldsymbol{\theta}}_u(k)^\top \; 1\right]^\top$$

according to Eq. (A2). Moreover, using (A1), the following equality holds:

$$\bar{\boldsymbol{q}}_u(k)^{-1}\tilde{\boldsymbol{q}}_0(k-1|k-1)\bar{\boldsymbol{q}}_u(k) = \begin{bmatrix} \boldsymbol{R}\left(\bar{\boldsymbol{q}}_u(k)\right)^\top \frac{1}{2}\tilde{\boldsymbol{\theta}}_0(k-1|k-1)^\top \\ 1 \end{bmatrix}$$

Finally, recalling that in (B2) we are multiplying small rotations, and using (A3), we can now write a first-order approximation of the orientation error vector as:

$$\tilde{\boldsymbol{\theta}}_0(k|k-1) \approx \boldsymbol{R}\left(\bar{\boldsymbol{q}}_u(k)\right)^\top \tilde{\boldsymbol{\theta}}_0(k-1|k-1) + \tilde{\boldsymbol{\theta}}_u(k) \tag{B3}$$

Equation (B3) constitutes a first-order expression of the error of the orientation estimate after prediction. For deriving the error state equations we also need to compute a first-order expression of the Cartesian error. In order to compute $\tilde{\boldsymbol{p}}_0(k|k-1)$ we can subtract memberwise the first equations in (1) and (2).

If, for notational simplification, we define

$$\begin{gathered} \boldsymbol{p}_u^r(k) = \boldsymbol{q}_0(k-1)\boldsymbol{p}_u(k)\boldsymbol{q}_0(k-1)^{-1} \\ \text{and} \\ \bar{\boldsymbol{p}}_u^r(k) = \hat{\boldsymbol{q}}_0(k-1|k-1)\bar{\boldsymbol{p}}_u(k)\hat{\boldsymbol{q}}_0(k-1|k-1)^{-1} \end{gathered}$$

we can write:

$$\boldsymbol{p}_0(k) \ominus \hat{\boldsymbol{p}}_0(k|k-1) = \boldsymbol{p}_0(k-1) \oplus \boldsymbol{p}_u^r(k) \ominus \hat{\boldsymbol{p}}_0(k-1|k-1) \ominus \bar{\boldsymbol{p}}_u^r(k) \tag{B4}$$

Now, remembering that $\tilde{\boldsymbol{p}}_0(k|k-1) = \boldsymbol{p}_0(k) \ominus \hat{\boldsymbol{p}}_0(k|k-1)$ and $\tilde{\boldsymbol{p}}_0(k-1|k-1) = \boldsymbol{p}_0(k-1) \ominus \hat{\boldsymbol{p}}_0(k-1|k-1)$ are the estimation error of the prior at time k and the estimation error of the posterior at time $k-1$, and rewriting $\boldsymbol{q}_0(k-1)$ and $\boldsymbol{p}_u(k)$ in function of the corresponding estimated (measured) quantities, we can develop expression (B4) as follows:

$$\begin{aligned} \tilde{\boldsymbol{p}}_0(k|k-1) &= \tilde{\boldsymbol{p}}_0(k-1|k-1) \oplus \boldsymbol{p}_u^r(k) \ominus \bar{\boldsymbol{p}}_u^r(k) \\ &= \tilde{\boldsymbol{p}}_0(k-1|k-1) \oplus (\hat{\boldsymbol{q}}_0(k-1|k-1)\tilde{\boldsymbol{q}}_0(k-1|k-1))\,(\bar{\boldsymbol{p}}_u(k) \oplus \tilde{\boldsymbol{p}}_u(k)) \\ &\quad \tilde{\boldsymbol{q}}_0(k-1|k-1)^{-1}\hat{\boldsymbol{q}}_0(k-1|k-1)^{-1} \ominus \bar{\boldsymbol{p}}_u^r(k) \end{aligned}$$

Rewriting the homogeneous coordinates in terms of 3D vectors and according to (A1), the previous expression becomes:

$$\begin{aligned} \tilde{\boldsymbol{t}}_0(k|k-1) = \tilde{\boldsymbol{t}}_0(k-1|k-1) &\\ + \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))\boldsymbol{R}(\tilde{\boldsymbol{q}}_0(k-1|k-1))\bar{\boldsymbol{t}}_u(k) &\\ + \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))\boldsymbol{R}(\tilde{\boldsymbol{q}}_0(k-1|k-1))\tilde{\boldsymbol{t}}_u(k) &\\ - \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))\bar{\boldsymbol{t}}_u(k) & \end{aligned}$$

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

We can now consider the small-angle approximation (A4) for the rotation matrix $\boldsymbol{R}(\tilde{\boldsymbol{q}}_0(k-1|k-1))$, and, neglecting higher-order terms, we obtain:

$$\begin{aligned}\tilde{\boldsymbol{t}}_0(k|k-1) \approx\ & \tilde{\boldsymbol{t}}_0(k-1|k-1) \\ & + \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))\left(\boldsymbol{I}_3 + \boldsymbol{S}(\tilde{\boldsymbol{\theta}}_0(k-1|k-1))\right)\bar{\boldsymbol{t}}_u(k) \\ & + \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))\left(\boldsymbol{I}_3 + \boldsymbol{S}(\tilde{\boldsymbol{\theta}}_0(k-1|k-1))\right)\tilde{\boldsymbol{t}}_u(k) \\ & - \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))\bar{\boldsymbol{t}}_u(k) \\ =\ & \tilde{\boldsymbol{t}}(k-1|k-1) \\ & + \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))\boldsymbol{S}(\tilde{\boldsymbol{\theta}}_0(k-1|k-1))\bar{\boldsymbol{t}}_u(k) \\ & + \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))\tilde{\boldsymbol{t}}_u(k)\end{aligned}$$

The previous expression can be further simplified by recalling that for generic vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$ it holds $\boldsymbol{S}(\boldsymbol{a})\boldsymbol{b} = -\boldsymbol{S}(\boldsymbol{b})\boldsymbol{a}$:

$$\begin{aligned}\tilde{\boldsymbol{t}}_0(k|k-1) = & \tilde{\boldsymbol{t}}_0(k-1|k-1) \\ & - \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))\boldsymbol{S}(\bar{\boldsymbol{t}}_u(k))\tilde{\boldsymbol{\theta}}_0(k-1|k-1) \\ & + \boldsymbol{R}(\hat{\boldsymbol{q}}_0(k-1|k-1))\tilde{\boldsymbol{t}}_u(k)\end{aligned} \tag{B5}$$

Reporting Eqs. (B3) and (B5) in matrix form we obtain:

$$\tilde{\boldsymbol{x}}_0(k|k-1) = \boldsymbol{F}_k\tilde{\boldsymbol{x}}_0(k|k-1) + \boldsymbol{G}_k\tilde{\boldsymbol{u}}(k)$$

with $\boldsymbol{F}_k$ and $\boldsymbol{G}_k$ defined as in (3).

Appendix C: Innovation for the j-th Measurement

In this section we derive the first-order expression (8) of the innovation for the j-th measurement. Let us consider the second equation in model (7) and rewrite the true quantities in terms of estimates (priors) and residual errors:

$$\begin{aligned}\bar{\boldsymbol{q}}_{z_j}(k) = & (\hat{\boldsymbol{q}}_0(k|k-1)\tilde{\boldsymbol{q}}_0(k|k-1))^{-1} \\ & \left(\hat{\boldsymbol{q}}_j(k|k-1)\tilde{\boldsymbol{q}}_j(k|k-1)\right)\tilde{\boldsymbol{q}}_{z_j}(k)^{-1}\end{aligned} \tag{C1}$$

Multiplying each member of (C1) by the quantity $\hat{\boldsymbol{q}}_{z_j}(k)^{-1} = \left(\hat{\boldsymbol{q}}_0(k|k-1)^{-1}\hat{\boldsymbol{q}}_j(k|k-1)\right)^{-1}$ we obtain:

$$\begin{aligned}\hat{\boldsymbol{q}}_{z_j}^{-1}(k)\bar{\boldsymbol{q}}_{z_j}(k) = & \left(\hat{\boldsymbol{q}}_0(k|k-1)^{-1}\hat{\boldsymbol{q}}_j(k|k-1)\right)^{-1} \\ & \tilde{\boldsymbol{q}}_0(k|k-1)^{-1}\left(\hat{\boldsymbol{q}}_0(k|k-1)^{-1}\hat{\boldsymbol{q}}_j(k|k-1)\right) \\ & \tilde{\boldsymbol{q}}_j(k|k-1)\tilde{\boldsymbol{q}}_{z_j}(k)^{-1}\end{aligned} \tag{C2}$$

But, by definition, the quantity $\hat{\boldsymbol{q}}_{z_j}^{-1}(k)\bar{\boldsymbol{q}}_{z_j}(k)$ is the *innovation* of the relative orientation measurement between the robot and landmark j, namely $\tilde{\boldsymbol{q}}_{r_j}(k)$; moreover, by applying (A1) and (A2), a first-order approximation of (C2) can be computed as:

$$\begin{aligned}\tilde{\boldsymbol{\theta}}_{r_j}(k) \approx & -\boldsymbol{R}\left(\hat{\boldsymbol{q}}_0(k|k-1)^{-1}\hat{\boldsymbol{q}}_j(k|k-1)\right)^{\top}\tilde{\boldsymbol{\theta}}_0(k|k-1) \\ & + \tilde{\boldsymbol{\theta}}_j(k|k-1) - \tilde{\boldsymbol{\theta}}_{z_j}(k) \qquad \forall j \in \mathcal{M}\end{aligned} \tag{C3}$$

The previous expression provides a first-order relation between the *innovation*, the prior error and the measurement error, for the orientation variables. We can further derive the innovation for the

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

Cartesian variables, starting from the first equation in (7):

$$\begin{aligned}\bar{\boldsymbol{p}}_{z_j}(k) = &(\hat{\boldsymbol{q}}_0(k|k-1)\tilde{\boldsymbol{q}}_0(k|k-1))^{-1} \\ &\left(\hat{\boldsymbol{p}}_j(k|k-1) \ominus \hat{\boldsymbol{p}}_0(k|k-1) \oplus \tilde{\boldsymbol{p}}_j(k|k-1) \ominus \tilde{\boldsymbol{p}}_0(k|k-1)\right) \\ &(\hat{\boldsymbol{q}}_0(k|k-1)\tilde{\boldsymbol{q}}_0(k|k-1)) \ominus \tilde{\boldsymbol{p}}_{z_j}(k)\end{aligned}$$

Using (A1) and passing to 3D coordinates for position variables, we can write the quaternions in terms of the corresponding rotation matrices:

$$\begin{aligned}\bar{\boldsymbol{t}}_{z_j}(k) = &-\tilde{\boldsymbol{t}}_{z_j}(k) + \boldsymbol{R}^\top(\tilde{\boldsymbol{q}}_0(k|k-1))\,\boldsymbol{R}^\top(\hat{\boldsymbol{q}}_0(k|k-1)) \\ &\left(\hat{\boldsymbol{t}}_j(k|k-1) - \hat{\boldsymbol{t}}_0(k|k-1) + \tilde{\boldsymbol{t}}_j(k|k-1) - \tilde{\boldsymbol{t}}_0(k|k-1)\right)\end{aligned}$$

Taking a small-angle approximation for the rotation matrix $\boldsymbol{R}(\tilde{\boldsymbol{q}}_0(k|k-1))^\top$, we obtain:

$$\begin{aligned}\bar{\boldsymbol{t}}_{z_j}(k) \approx &-\tilde{\boldsymbol{t}}_{z_j}(k) + \left(\boldsymbol{I}_3 + \boldsymbol{S}^\top\left(\tilde{\boldsymbol{\theta}}_0(k|k-1)\right)\right)\boldsymbol{R}^\top(\hat{\boldsymbol{q}}_0(k|k-1)) \\ &\left(\hat{\boldsymbol{t}}_j(k|k-1) - \hat{\boldsymbol{t}}_0(k|k-1) + \tilde{\boldsymbol{t}}_j(k|k-1) - \tilde{\boldsymbol{t}}_0(k|k-1)\right)\end{aligned}$$

After few matrix multiplications and neglecting higher order terms in the residual errors we can write:

$$\begin{aligned}\bar{\boldsymbol{t}}_{z_j}(k) \approx &-\tilde{\boldsymbol{t}}_{z_j}(k) + \boldsymbol{R}^\top(\hat{\boldsymbol{q}}_0(k|k-1))\left(\tilde{\boldsymbol{t}}_j(k|k-1) - \tilde{\boldsymbol{t}}_0(k|k-1)\right) \\ &- \boldsymbol{S}\left(\tilde{\boldsymbol{\theta}}_0(k|k-1)\right)\boldsymbol{R}^\top(\hat{\boldsymbol{q}}_0(k|k-1))\left(\hat{\boldsymbol{t}}_j(k|k-1) - \hat{\boldsymbol{t}}_0(k|k-1)\right) \\ &+ \boldsymbol{R}^\top(\hat{\boldsymbol{q}}_0(k|k-1))\left(\hat{\boldsymbol{t}}_j(k|k-1) - \hat{\boldsymbol{t}}_0(k|k-1)\right)\end{aligned} \tag{C4}$$

Now, revising the expression for the Cartesian measurement in (6), it is possible to see that the last summand in (C4) is the expected measurement, given the current state estimate, namely $\hat{\boldsymbol{t}}_{z_j}(k)$, hence we can write:

$$\begin{aligned}\bar{\boldsymbol{t}}_{z_j}(k) - \hat{\boldsymbol{t}}_{z_j}(k) \approx &-\tilde{\boldsymbol{t}}_{z_j}(k) + \boldsymbol{R}^\top(\hat{\boldsymbol{q}}_0(k|k-1))\left(\tilde{\boldsymbol{t}}_j(k|k-1) - \tilde{\boldsymbol{t}}_0(k|k-1)\right) \\ &-\boldsymbol{S}(\tilde{\boldsymbol{q}}_0(k|k-1))\,\boldsymbol{R}^\top(\hat{\boldsymbol{q}}_0(k|k-1))\left(\hat{\boldsymbol{t}}_j(k|k-1) - \hat{\boldsymbol{t}}_0(k|k-1)\right)\end{aligned}$$

By definition, $\tilde{\boldsymbol{t}}_{r_j}(k) \doteq \bar{\boldsymbol{t}}_{z_j}(k) - \hat{\boldsymbol{t}}_{z_j}(k)$ is the *innovation* for the relative position measurement between the robot and landmark j. Hence, we can finally derive the expression of the innovation in function of the Cartesian measurement and estimation errors:

$$\begin{aligned}\tilde{\boldsymbol{t}}_{r_j}(k) \approx &-\tilde{\boldsymbol{t}}_{z_j}(k) + \boldsymbol{R}^\top(\hat{\boldsymbol{q}}_0(k|k-1))\left(\tilde{\boldsymbol{t}}_j(k|k-1) - \tilde{\boldsymbol{t}}_0(k|k-1)\right) \\ &+ \boldsymbol{S}\left(\boldsymbol{R}^\top(\hat{\boldsymbol{q}}_0(k|k-1))\left(\hat{\boldsymbol{t}}_j(k|k-1) - \hat{\boldsymbol{t}}_0(k|k-1)\right)\right)\tilde{\boldsymbol{\theta}}_0(k|k-1)\end{aligned} \tag{C5}$$

From Eqs. (C3) and (C5) the innovation vector $\tilde{\boldsymbol{r}}_j = [\tilde{\boldsymbol{t}}_{r_j}^\top(k)\ \tilde{\boldsymbol{\theta}}_{r_j}^\top(k)]^\top$ for measurement from the j-th landmark can be written in compact form as:

$$\tilde{\boldsymbol{r}}_j(k) \approx \boldsymbol{H}_j\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{z}}_j$$

with $\boldsymbol{H}_j$ defined as in (9).

Appendix D: Registration of Uncertain 3D Vectors

Let us consider two sets of vectors $\boldsymbol{A} = \{\boldsymbol{a}_i \in \mathbb{R}^3, i = 1, \ldots, \kappa\}$ and $\boldsymbol{B} = \{\boldsymbol{b}_i \in \mathbb{R}^3, i = 1, \ldots, \kappa\}$, each one describing a collection of κ features (points, line directions or normals to planes) observed from a given vantage pose, respectively $\mathcal{R}_A$ and $\mathcal{R}_B$. The point features are described by κ_p 3D vectors, while normals and directions correspond to κ_u three-dimensional *unit vectors*, with $\kappa = \kappa_p + \kappa_u$. The *vector registration problem* consists in finding the roto-translation $(\boldsymbol{R}, \boldsymbol{t})$ that relates the reference

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

frames $\mathcal{R}_A$ and $\mathcal{R}_B$, given $\boldsymbol{A}$ and $\boldsymbol{B}$. In this appendix we will briefly present an approach for computing the roto-translation achieving the optimal match between the given vector sets. This technique is based on a Gauss-Newton method and the interested reader can find a detailed study on registration techniques in ref. [11] and [12]. It is possible to rewrite the measured vectors as $\boldsymbol{a}_i = \boldsymbol{\alpha}_i - \boldsymbol{\varepsilon}_{a_i}$ and $\boldsymbol{b}_i = \boldsymbol{\beta}_i - \boldsymbol{\varepsilon}_{b_i}$, $i = 1, \ldots, \kappa$, where $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ are the true vectors, whereas the terms $\boldsymbol{\varepsilon}_{a_i}$ and $\boldsymbol{\varepsilon}_{b_i}$ are zero mean independent Gaussian noises, i.e., $\boldsymbol{\varepsilon}_{a_i} \sim \mathcal{N}(\boldsymbol{0}_3, \boldsymbol{P}_i^a)$ and $\boldsymbol{\varepsilon}_{b_i} \sim \mathcal{N}(\boldsymbol{0}_3, \boldsymbol{P}_i^b)$, being $\boldsymbol{P}_i^a$ and $\boldsymbol{P}_i^b$ the 3 by 3 covariance matrices associated to the vectors $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$, respectively. Under this setup the vector registration problem becomes a minimization of the following Mahalanobis distance:

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{R}, \boldsymbol{t}) = &\sum_{i=1}^{\kappa_p} (\boldsymbol{a}_i - \boldsymbol{R}\boldsymbol{b}_i - \boldsymbol{t})^\top \boldsymbol{P}_i^{-1} (\boldsymbol{a}_i - \boldsymbol{R}\boldsymbol{b}_i - \boldsymbol{t}) \\ &+ \sum_{i=\kappa_p+1}^{\kappa} (\boldsymbol{a}_i - \boldsymbol{R}\boldsymbol{b}_i)^\top \boldsymbol{P}_i^{-1} (\boldsymbol{a}_i - \boldsymbol{R}\boldsymbol{b}_i), \end{aligned} \tag{D1}$$

where $\boldsymbol{P}_i = \boldsymbol{P}_i^a + \boldsymbol{R}\boldsymbol{P}_i^b\boldsymbol{R}^\top$ is the *registration covariance* expressed in the reference frame $\mathcal{R}_A$.

Let us consider the expression of a single residual in the previous summand:

$$\boldsymbol{\xi}_i = \boldsymbol{a}_i - \boldsymbol{R}\boldsymbol{b}_i - \lambda_i \boldsymbol{t}, \tag{D2}$$

where the support variable λ_i is equal to 1 for $i = 1, \ldots, \kappa_p$ and equal to zero for the other vectors. Let us assume that an initial guess $\bar{\boldsymbol{R}}$ for optimization is available and let us express the rotation $\boldsymbol{R}$ in the residual (D2) as a function of such known rotation and an unknown correction $\boldsymbol{R}(\boldsymbol{\delta}_\theta)$:

$$\boldsymbol{\xi}_i = \boldsymbol{a}_i - \bar{\boldsymbol{R}}\boldsymbol{R}(\boldsymbol{\delta}_\theta)\boldsymbol{b}_i - \lambda_i \boldsymbol{t}. \tag{D3}$$

Assuming the correction angle $\boldsymbol{\delta}_\theta$ to be small we can approximate $\boldsymbol{R}(\boldsymbol{\delta}_\theta) = \boldsymbol{I}_3 + \boldsymbol{S}(\boldsymbol{\delta}_\theta)$, hence (D3) becomes:

$$\boldsymbol{\xi}_i \approx \boldsymbol{a}_i - \bar{\boldsymbol{R}}\boldsymbol{b}_i - \bar{\boldsymbol{R}}\boldsymbol{S}(\boldsymbol{\delta}_\theta)\boldsymbol{b}_i - \lambda_i \boldsymbol{t} = \boldsymbol{a}_i - \bar{\boldsymbol{R}}\boldsymbol{b}_i - \lambda_i \boldsymbol{t} + \bar{\boldsymbol{R}}\boldsymbol{S}(\boldsymbol{b}_i)\boldsymbol{\delta}_\theta = \boldsymbol{a}_i - \bar{\boldsymbol{R}}\boldsymbol{b}_i - \boldsymbol{\Lambda}_i \boldsymbol{\delta},$$

with $\boldsymbol{\Lambda}_i = [\lambda_i \boldsymbol{I}_3 \; - \bar{\boldsymbol{R}}\boldsymbol{S}(\boldsymbol{b}_i)]$, $i = 1, \ldots, \kappa$ and $\boldsymbol{\delta} = [\boldsymbol{t}^\top \; \boldsymbol{\delta}_{\boldsymbol{\theta}}^\top]^\top$. We can now rewrite a quadratic approximation of the objective function (D1) around the initial guess $\bar{\boldsymbol{R}}$ as:

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{R}, \boldsymbol{t}) &= \sum_{i=1}^{\kappa} \boldsymbol{\xi}_i^\top \boldsymbol{P}_i^{-1} \boldsymbol{\xi}_i \\ &\approx \sum_{i=1}^{\kappa} \left(\boldsymbol{a}_i - \bar{\boldsymbol{R}}\boldsymbol{b}_i - \boldsymbol{\Lambda}_i \boldsymbol{\delta}\right)^\top \bar{\boldsymbol{P}}_i^{-1} \left(\boldsymbol{a}_i - \bar{\boldsymbol{R}}\boldsymbol{b}_i - \boldsymbol{\Lambda}_i \boldsymbol{\delta}\right). \end{aligned}$$

with $\bar{\boldsymbol{P}}_i = \boldsymbol{P}_i^a + \bar{\boldsymbol{R}}\boldsymbol{P}_i^b\bar{\boldsymbol{R}}^\top$. The solution to the previous convex optimization problem can be computed as:[5]

$$\boldsymbol{\delta}^* = \left(\sum_{j=1}^{\kappa} \boldsymbol{\Lambda}_j^\top \bar{\boldsymbol{P}}_j^{-1} \Lambda_j \right)^{-1} \sum_{i=1}^{\kappa} \boldsymbol{\Lambda}_i^\top \bar{\boldsymbol{P}}_i^{-1} (\boldsymbol{a}_i - \bar{\boldsymbol{R}}\boldsymbol{b}_i).$$

The previous solution provides an estimate of the translation $\boldsymbol{t}^*$ and a local angular correction $\boldsymbol{\delta}_\theta^*$. The algorithm can be iterated until convergence, using the refined rotation matrix estimate $\bar{\boldsymbol{R}}\boldsymbol{R}(\boldsymbol{\delta}_\theta^*)$ as initial guess for a new iteration.

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press

According to the framework presented so far it is also possible to devise a covariance matrix of the estimated roto-translation:

$$\boldsymbol{P}^* = \left(\sum_{j=1}^{\kappa} \boldsymbol{\Lambda}_j^\top \bar{\boldsymbol{P}}_j^{-1} \boldsymbol{\Lambda}_j \right)^{-1} . \tag{D4}$$

For sake of brevity we refer the reader to[12] for the derivation of $\boldsymbol{P}^*$. In the same reference it is shown that the registration algorithm described so far, with a suitable initialization of the iterative nonlinear optimization method, converges to a minimum of the objective function in few iterations and assures that the estimation errors are consistent with the covariance matrix (D4). Finally, we observe that when the quality of the data association is expected to be poor, the registration approach can be integrated within a `RANSAC` algorithm.

https://doi.org/10.1017/S0263574714000678 Published online by Cambridge University Press