


Online self-calibration for robotic systems

Jérôme Maye, Hannes Sommer, Gabriel Agamennoni,
Roland Siegwart and Paul Furgale

The International Journal of
Robotics Research
2016, Vol. 35(4) 357–380
© The Author(s) 2015
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364915596232
ijr.sagepub.com


Abstract

We present a generic algorithm for self-calibration of robotic systems that utilizes two key innovations. First, it uses an information-theoretic measure to automatically identify and store novel measurement sequences. This keeps the computation tractable by discarding redundant information and allows the system to build a sparse but complete calibration data-set from data collected at different times. Second, as the full observability of the calibration parameters may not be guaranteed for an arbitrary measurement sequence, the algorithm detects and locks unobservable directions in parameter space using a combination of rank-revealing QR and singular value decompositions of the Fisher information matrix. The result is an algorithm that listens to an incoming sensor stream, builds a minimal set of data for estimating the calibration parameters, and updates parameters as they become observable, leaving the others locked at their initial guess. We validate our approach through an extensive set of simulated and real-world experiments.

Keywords

Online calibration, observability, batch-estimation, Gauss-Newton

1. Introduction

Every robotic system has some set of parameters, scale factors, sensor locations, link lengths, etc., that are needed for state estimation, planning, and control. Despite best efforts during construction, some parameters will change over the lifetime of a robot due to normal wear and tear. In the best case, incorrect parameter values degrade performance. In the worst case, they cause critical safety issues. We are interested in developing automated systems that are capable of robust and long-term deployment in the hands of non-experts, so the automatic identification and update of these parameter values is highly important.

As an example, consider a camera-based collision avoidance system intended for deployment in a consumer automobile. For the vehicle to be able to avoid collisions, the pose of each camera with respect to the vehicle coordinate system must be known precisely so that obstacle positions can be transformed from camera coordinates into vehicle coordinates. However, a consumer vehicle will have no access to special calibration hardware or expert data analysis. In such a scenario, the vehicle must be capable of self-recalibration. This problem is inherently difficult for a number of reasons that we will briefly discuss here.

- (1) **Parameters change over time:** although the vehicle may be factory calibrated, the parameters can change

slowly over time due to vibration, thermal expansion, loose parts, or any number of other common problems that follow from normal usage.

- (2) **Parameters must be inferred from the data:** as the cameras may be installed in different places on the vehicle, their pose cannot be measured directly. Instead, the parameters must be inferred from the data produced by the full system. However, this is only possible if the motion of the vehicle renders the parameters *observable*.¹
- (3) **Normal operation may result in unobservable directions in parameter space:** unfortunately, for this and many other practical problems normal operation may not render all directions in parameters space observable. In this example, when two cameras do not share an overlapping field of view, planar motion renders the calibration problem degenerate;² the transformation between cameras only becomes observable under general 3D motion.

Autonomous Systems Lab, ETH Zurich, Switzerland

Corresponding author:

Hannes Sommer, Autonomous Systems Lab, ETH Zurich, Institute of Robotics and Intelligent Systems, Leonhardstrasse 21, Zurich 8092, Switzerland.
Email: hannes.sommer@mavt.ethz.ch

- (4) **Unobservable directions in parameter space may appear observable in the presence of noise:** even if our hypothetical car is piloted only on a plane (a degenerate case), noise in the measurements can make unobservable parameters appear observable. We call these parameters *numerically unobservable* to mirror the related concept of numerically rank-deficient matrices.

Existing algorithms for self-calibration generally handle issues (1) and (2). Issue (3) is dealt with by designing experiments that guarantee all parameters become observable. Issue (4) has been largely ignored. Therefore, unless we plan to require all vehicle owners to outfit their parking places with calibration patterns or regularly drive off-road to make the parameters observable, new advances for online system calibration are required. To the best of our knowledge, our previous conference paper (that this paper extends) (Maye et al., 2013) is the first published self-calibration algorithm able to cope with issue (4).

In this paper, we propose an algorithm to deal with *all* of the aforementioned difficulties. Our approach exploits the algebraic links between the Gauss–Newton algorithm, the Fisher information matrix, and nonlinear observability analysis to automatically detect directions in parameter space that are numerically unobservable and avoid updating our parameters in these directions; at any given time, directions in the parameter space that are observable will be updated based on the latest information, while unobservable directions will remain at their initial guess. Novel sets of measurements are detected using an information gain test and added to a working set that is used to estimate the parameters. The result is an algorithm that listens to an incoming stream of data, automatically accumulating a batch of data that may be used to calibrate a full robot system. The only requirements are that (i) the parameters are theoretically observable given some ideal set of data, (ii) it is possible to implement a batch Gauss–Newton estimator for the system state and calibration parameters based on a set of measurements, and (iii) we have some reasonable initial guess for the calibration parameters (e.g. from factory calibration or manual measurements).

This paper follows on from our previous conference paper (Maye et al., 2013). The method and the experiments have both been extended significantly. The remainder of the paper is structured as follows. Section 2 will give a brief overview over related approaches. Section 3 is dedicated to the mathematical groundings of our method. Section 4 demonstrates the validity of our approach through extensive experiments and evaluation. Section 5 will conclude the paper.

2. Related work

The problem of sensor calibration has been a recurring one in the history of robotics and computer vision. Thereby, it

has been addressed using a variety of sensor setups and algorithms. A calibration process may involve recovering *intrinsic*, e.g. focal length for a camera, and *extrinsic* parameters, i.e. the rigid transformation between the sensor's coordinate system and a reference coordinate system. For the former, one could devise a naive approach where the parameters are accurately determined during the manufacturing process. Similarly, the transformation could be retrieved by means of some measuring instrument. However, the disadvantages of such purely engineered methods are manifold. Apart from their impracticality, it can be nearly impossible to reach a satisfying level of accuracy and thus hinder the proper use of the sensor in a robotic system. Furthermore, external factors such as temperature variations or mechanical shocks may seriously bias a factory calibration. Therefore, much efforts have been dedicated over the years to develop algorithms for calibration of systems in the field.

At a first level, sensor calibration methods can be classified into online or offline algorithms. While the latter are more popular in the computer vision community, the former are widespread in robotic applications. Online methods usually augment the state vector of a Bayes filter with calibration parameters and offline approaches rely on the minimization of a cost function. At a second level, calibration algorithms can be distinguished in how they address observability or singularity issues. While some publications present an informal or formal offline analysis, some others silently assume that the dataset contains enough information to calibrate the sensor parameters. Finally, the complexity of the setup and the amount of supervision are substantial criteria to assess the practicality of a calibration method.

Due to its simplicity, the use of a planar calibration pattern such as a checkerboard, coupled with nonlinear regression, has become the most popular method in computer vision during the last decade for calibrating intrinsic camera parameters (Sturm and Maybank, 1999; Zhang, 1999). Given a set of point correspondences between the model planes and their images, the algorithm attempts to minimize a nonlinear cost function consisting in the sum of squared reprojection errors. While Zhang (1999) remained rather vague about degenerate configurations, a compendium of singularities is proposed by Sturm and Maybank (1999). The planar calibration pattern is also used by Zhang and Pless (2004) to estimate the relative pose between a camera and a laser rangefinder (LRF), with no particular attention to singularities in the camera views. While being relatively efficient and inexpensive, this procedure still requires expert knowledge to reach a good level of accuracy. Even though point correspondences can be automatically identified by corner detectors (Harris and Stephens, 1988; Rufli et al., 2008), it is still up to the user to execute proper motion of the calibration pattern in front of the camera.

A key limitation of the plane-based calibration technique for intrinsic camera parameters is that it remains essentially offline. Whenever a change of calibration occurs, e.g. an

adjustment of the focal length, the vision task has to be interrupted for recalibration. In order to allow for online operation, Maybank and Faugeras (1992) have pioneered a method for self-calibration that does not require any pattern with known 3D geometry. Given a couple of images from the same scene under different viewpoints and point correspondences, epipolar transformations are computed and a system of equations is solved to yield the camera intrinsic parameters. Due to the lack of probabilistic model, the method is inherently sensitive to noise. Furthermore, a proper calibration cannot be guaranteed under general camera motion.

In the context of mobile robotics, several authors have included the calibration problem in a state estimation framework, either with filtering (Gao and Spletzer, 2010; Kelly and Sukhatme, 2011; Martinelli et al., 2006; Mirzaei and Roumeliotis, 2008) or nonlinear least-squares (Censi et al., 2008; Heng et al., 2013; Kuemmerle et al., 2011) techniques. Filtering techniques based on the Kalman filter are appealing due to their inherently online nature. However, in case of nonlinear systems, least-squares techniques based on iterative optimization are usually superior in terms of accuracy. Martinelli et al. (2006) perform a nonlinear observability analysis using Lie derivatives and overcome the singularities by choosing an alternative parameterization for the robot's state. An identical method for observability analysis has also been employed by Mirzaei and Roumeliotis (2008) and Kelly and Sukhatme (2011).

More recently, Brookshire and Teller (2011, 2012) have carried out a formal observability analysis in order to identify degenerate paths of the calibration dataset. In contrast to the aforementioned methods, they inspect the rank of the Fisher information matrix to determine the observability of the calibration parameters. However, their approach still expects that the robot travels along non-degenerate paths during the calibration run. A standard nonlinear least-squares method is subsequently used for the estimation.

Several authors have proposed including the estimation of time offsets between sensor clocks into calibration problems for use online (Li and Mourikis, 2014) and offline (Furgale et al., 2013a; Kelly et al., 2014; Mair et al., 2011; Rehder et al., 2014). Li and Mourikis (2014) in particular carefully enumerated the cases that render the temporal offset between a camera and IMU to become unobservable. As with the work described above, none of these approaches proposes a method to handle these unobservable cases. Other authors have developed tools for estimating and compensating for clock skew and offset in a sensor network, based on measurements of network delay (Harrison and Newman, 2011; Zhang et al., 2002). These methods specifically address the temporal calibration problem, but are not designed for estimating intrinsic or extrinsic sensor calibration parameters.

Another class of methods seeks to maximize the concordance of the sensor data with the environment (Levinson and Thrun, 2010; Underwood et al., 2007) or a quality measure (Maddern et al., 2012; Sheehan et al., 2010, 2012).

While Underwood et al. (2007) use a vertical pole with retro reflective tape, Levinson and Thrun (2010) propose an unsupervised approach relying on the assumption that neighboring points from laser beams lie on a contiguous surface. The method presented by Sheehan et al. (2010), Maddern et al. (2012), and Sheehan et al. (2012) estimates relative poses of LRFs by minimizing the entropy of an underlying generative model for the location of laser points in space, hence maximizing the sharpness of the distribution and the quality of the resulting point cloud. Along similar lines, Teichman et al. (2013) presented an approach for estimating the intrinsic calibration parameters of a depth sensor. The approach defines a generic objective function that penalizes range-dependent depth distortions. The algorithm alternates between calibration and simultaneous localization and mapping (SLAM) optimization, and adopts different objective functions for each step. Hence it is not guaranteed to converge.

Closely related to our approach, Menq et al. (1989) have also adopted matrix factorization to identify observable and unobservable subspaces in the context of calibration of industrial robots. However, the authors do not explicitly deal with the notion of numerical rank deficiency, which, in our opinion, is essential to the proper operation of a calibration algorithm.

To the best of the authors' knowledge, little research has been devoted to account for degenerate cases frequently occurring during calibration runs. Some authors, notably in the robotics community, have proposed an offline observability analysis to either state that the system is observable or to identify potential singularities. As will be demonstrated in this paper, traditional observability analysis is helpless when it comes to noisy input data on real physical systems. It is indeed purely algebraic, performed a priori, and hence omits the numerical difficulties associated with sensory data. In general, apart from Menq et al. (1989), the authors assume that their estimation framework is fed with well-behaved data. In contrast, our approach relaxes this assumption and is therefore suitable in the hands of non-expert users for online and long-term operations on various platforms and sensors.

3. Method

In this section, we shall first state a probabilistic model for the calibration problem and present an in-depth description of our inference method. In a second step, we shall propose an online extension to our approach. Practical algorithmic considerations will conclude the presentation.

3.1. Probabilistic model

The first step of our calibration method is to identify the different variables in the system and establish a probabilistic model through which they interact. These variables shall be classified in the following three categories.

- (1) The data variable denoted by $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_D)^T$ and samples thereof as \mathbf{X}_i . A realization of the observable

random variable, \mathbf{X}_i , corresponds to the measurement of a sensor, \mathcal{S} .

- (2) The calibration variable denoted by $\Theta = (\Theta_1, \dots, \Theta_K)^T$. This latent random variable is the primary object of interest and comprises all the intrinsic and extrinsic calibration parameters of the sensor \mathcal{S} .
- (3) The nuisance variable represented by $\Psi = (\Psi_1, \dots, \Psi_L)^T$. This latent random variable is not of direct interest, but the data distribution depends on it (e.g. poses of landmarks).

For the sake of clarity, we restrict our formulation to a single sensor \mathcal{S} . Nevertheless, our approach smoothly generalizes to multiple sensors. We denote random variates of \mathbf{X} by \mathbf{x} , of Θ by θ , and of Ψ by ψ .

Assuming a realization $\mathbf{x}_1, \dots, \mathbf{x}_N$ of an i.i.d. data sample, henceforth denoted by $\mathbf{x}_{1:N}$, the posterior joint probability density function over the latent variables is

$$f_{\Psi, \Theta}(\psi, \theta | \mathbf{x}_{1:N}) = \frac{\prod_{i=1}^N f_{\mathbf{x}}(\mathbf{x}_i | \psi, \theta) f_{\Psi, \Theta}(\psi, \theta)}{\prod_{i=1}^N f_{\mathbf{x}}(\mathbf{x}_i)} \quad (1)$$

As our final goal is to perform inference on the calibration variable, Θ , we shall marginalize over the nuisance variable, Ψ , to obtain the marginal posterior density function

$$f_{\Theta}(\theta | \mathbf{x}_{1:N}) = \int f_{\Psi, \Theta}(\psi, \theta | \mathbf{x}_{1:N}) d\psi \quad (2)$$

From the large sample theory, the posterior density in Equation 1 can be approximated through Laplace's method (Gelman et al., 2003) by the normal distribution,

$$f_{\Psi, \Theta}(\psi, \theta | \mathbf{x}_{1:N}) \approx \mathcal{N}(\hat{\psi\theta}, \mathcal{I}^{-1}(\hat{\psi\theta})) \quad (3)$$

where $\hat{\psi\theta}$ is the joint posterior mode and $\mathcal{I}(\hat{\psi\theta})$ the observed Fisher information matrix, i.e. the negative of the Hessian matrix of the log-likelihood function evaluated at the joint posterior mode.

Using the properties of the normal distribution (Bishop, 2006), we can derive a closed-form expression for the posterior marginal density in Equation 2 as

$$f_{\Theta}(\theta | \mathbf{x}_{1:N}) \approx \mathcal{N}(\mu_{\Theta}, \Sigma_{\Theta}) \quad (4)$$

where μ_{Θ} and Σ_{Θ} represent partitions of the joint mean and covariance matrix in Equation 3, respectively.

To conclude this probabilistic model, we shall state the assumed form of the data distribution $f_{\mathbf{x}}$ in Equation 1 as

$$\mathbf{X} \sim \mathcal{N}(g_s(\psi, \theta), \Sigma_{\mathbf{X}}) \quad (5)$$

where $g_s(\cdot)$ is an arbitrary twice-differentiable function, henceforth assumed to be nonlinear, and $\Sigma_{\mathbf{X}}$ the deterministic covariance matrix of the measurements. The choice

of a normally distributed measurement model is standard in robotics. Borrowing from the inverse problem theory (Tarantola, 2005), we refer to the function $g_s(\cdot)$ as the forward model. It expresses the mathematical model of the physical system under study and predicts the outcome of \mathbf{X} for given model parameters ψ and θ .

3.2. Nonlinear least-squares problem

Under the assumptions of Section 3.1, the joint posterior mode in Equation 3 coincides with the solution to the least-squares problem

$$\hat{\psi\theta} = \underset{\psi, \theta}{\operatorname{argmin}} \sum_{i=1}^N \mathbf{e}_i^T \Sigma_{\mathbf{X}}^{-1} \mathbf{e}_i \quad (6)$$

where \mathbf{e}_i is an error term defined as

$$\mathbf{e}_i = \mathbf{x}_i - g_s(\psi, \theta) \quad (7)$$

We adopt an objective Bayesian viewpoint by setting an uninformative prior density $f_{\Psi, \Theta}(\psi, \theta) \propto 1$.

For the general case of a nonlinear function, $g_s(\cdot)$, Equation 6 has no closed-form solution and we have to resort to numerical optimization techniques (Kelley, 1999). In this paper, we adopt the Gauss–Newton algorithm that iteratively refines an initial guess, $\psi\theta_{(0)}$, towards a local minimum,

$$\psi\theta_{(t+1)} \leftarrow \psi\theta_{(t)} + \Delta\psi\theta_{(t)} \quad (8)$$

The shift vector, $\Delta\psi\theta_{(t)}$, is the solution to a linear least-squares problem constructed from a first-order linear approximation at the current estimate, $\psi\theta_{(t)}$. The least-squares estimate for the shift vector is thus the solution to the normal equations,

$$\mathbf{J}_{(t)}^T \Sigma^{-1} \mathbf{J}_{(t)} \Delta\psi\theta_{(t)} = \mathbf{J}_{(t)}^T \Sigma^{-1} \Delta\mathbf{e}_{(t)} \quad (9)$$

where $\mathbf{J}_{(t)}$ is the Jacobian matrix of the stacked error terms evaluated at the current estimate, Σ the covariance matrix of the whole data sample, and $\Delta\mathbf{e}_{(t)}$ the stacked vector of residuals at the current estimate.

Using the Cholesky decomposition of the positive-semidefinite inverse covariance matrix Σ^{-1} , we can rewrite Equation 9 as

$$\begin{aligned} & \left(\Sigma^{-\frac{T}{2}} \mathbf{J}_{(t)} \right)^T \left(\Sigma^{-\frac{T}{2}} \mathbf{J}_{(t)} \right) \Delta\psi\theta_{(t)} \\ &= - \left(\Sigma^{-\frac{T}{2}} \mathbf{J}_{(t)} \right)^T \left(\Sigma^{-\frac{T}{2}} \Delta\mathbf{e}_{(t)} \right) \end{aligned} \quad (10)$$

which, from linear algebra theory (Golub and Loan, 1996), can be recognized as the normal equations of the overdetermined system of linear equations,

$$\left(\Sigma^{-\frac{T}{2}}\mathbf{J}_{(t)}\right)\Delta\boldsymbol{\psi}\boldsymbol{\theta}_{(t)} = -\Sigma^{-\frac{T}{2}}\Delta\mathbf{e}_{(t)} \quad (11)$$

At convergence of the Gauss–Newton algorithm after I iterations, the parameter estimate becomes

$$\widehat{\boldsymbol{\psi}\boldsymbol{\theta}} = \boldsymbol{\psi}\boldsymbol{\theta}_{(I)} \quad (12)$$

and the observed Fisher information, constituting the inverse covariance matrix in Equation 3, is given by

$$\mathcal{I}(\widehat{\boldsymbol{\psi}\boldsymbol{\theta}}) = \mathbf{J}_{(I)}^T \Sigma^{-1} \mathbf{J}_{(I)} \quad (13)$$

Matrix factorization algorithms such as the singular value decomposition (SVD) or rank-revealing QR (RRQR) decomposition (Chan, 1987) provide a unique least-squares solution to Equation 11 whenever $\Sigma^{-T/2}\mathbf{J}_{(I)}$ has full rank.³ In case of rank deficiency, there are an infinite number of least-squares minimizers, the SVD providing the unique minimum-norm solution and RRQR decomposition a basic solution.

As the matrix rank is an algebraic concept, a matrix built from noisy sensor data or resulting from floating-point operations will rarely be truly rank deficient. Our algorithm relies on automated analysis of the rank of the Gauss–Newton system, even in the presence of noisy measurements. Fortunately, the concept of numerical rank (Golub and Loan, 1996) provides an elegant workaround to these difficulties. The numerical rank corresponds to the algebraic rank of the unperturbed matrix within a neighborhood defined by a parameter ϵ proportional to the magnitude of the perturbation matrix. When the noise on the matrix entries has the same scale (e.g. by column or row scaling), the numerical rank can be determined by the singular values (Hansen, 1998).

3.3. Parameter inference

This section explains our method of performing iterative Gauss–Newton-style estimation while automatically detecting and locking unobservable directions in the space of the calibration parameters, $\boldsymbol{\Theta}$. While $\boldsymbol{\Theta}$ is mostly low dimensional, $\boldsymbol{\Psi}$ can become high dimensional, especially in a SLAM scenario, where it encodes parameters for the robot states and the landmarks in the scene. Hence, the matrices involved in the Gauss–Newton system of equations are large, sparse, and may be rank deficient.

There are well-known methods to deal with the problem of rank deficiency in Gauss–Newton. The most widespread is the Levenberg–Marquardt (LM) algorithm that adds a scaled identity matrix to the left-hand side of the Gauss–Newton equations. This mutes the singular values of the Fischer information matrix and makes it possible to solve the linear system. However, this method is not suitable in our case; we are interested in performing automated rank

analysis of the reduced system so that we can determine the directions in parameter space that are numerically unobservable. LM changes the linear system and makes this analysis impossible.

The SVD would be the optimal tool for accurate numerical rank detection (Chan and Hansen, 1992) and matrix analysis. Unfortunately, this method is intractable on large sparse matrices, where we must use an RRQR decomposition. Therefore, we propose a method, inspired by the Schur complement, that exploits the benefits of each approach. The RRQR algorithm (suitable for large, sparse matrices) is used to marginalize out $\boldsymbol{\Psi}$. The remaining low-dimensional system of equations for $\boldsymbol{\Theta}$ can then be solved (and analyzed) using the SVD. Finally, back substitution is used to recover the shift vector corresponding to $\boldsymbol{\Psi}$. The method is unique in its partitioning of the problem and handling of rank deficient matrices.

In the Jacobian matrix of Equation 10, each column corresponds to one dimension of the parameter space. For our application, we shall thus partition it in the following manner

$$\mathbf{J} = (\mathbf{J}_{\boldsymbol{\psi}} \quad \mathbf{J}_{\boldsymbol{\theta}}) \quad (14)$$

where $\mathbf{J}_{\boldsymbol{\psi}} \in \mathbb{R}^{M \times L}$ corresponds to the Jacobian matrix with respect to $\boldsymbol{\psi}$, $\mathbf{J}_{\boldsymbol{\theta}} \in \mathbb{R}^{M \times K}$ with respect to $\boldsymbol{\theta}$, and $M = N \times D$. For the sake of clarity, we hence assume that the data covariance matrix, Σ , has been absorbed in the Jacobian matrix and the right-hand side of the normal equations using a Cholesky decomposition. Furthermore, we drop the minus sign and the iteration indexes. Given these assumptions, the normal equations in Equation 10 can be rewritten as

$$\begin{pmatrix} \mathbf{J}_{\boldsymbol{\psi}}^T \mathbf{J}_{\boldsymbol{\psi}} & \mathbf{J}_{\boldsymbol{\psi}}^T \mathbf{J}_{\boldsymbol{\theta}} \\ (\mathbf{J}_{\boldsymbol{\psi}}^T \mathbf{J}_{\boldsymbol{\theta}})^T & \mathbf{J}_{\boldsymbol{\theta}}^T \mathbf{J}_{\boldsymbol{\theta}} \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{\psi} \\ \Delta\boldsymbol{\theta} \end{pmatrix} = \begin{pmatrix} \mathbf{J}_{\boldsymbol{\psi}}^T \Delta\mathbf{e} \\ \mathbf{J}_{\boldsymbol{\theta}}^T \Delta\mathbf{e} \end{pmatrix} \quad (15)$$

3.3.1. Calibration variable estimation. In order to isolate the subsystem of equations related to $\Delta\boldsymbol{\theta}$ only, we need to cancel out the lower-left block of the Fisher information matrix by left-multiplying it with the full-rank matrix

$$\mathbf{C} = \begin{pmatrix} \mathbf{I}_{\boldsymbol{\psi}} & 0 \\ -(\mathbf{J}_{\boldsymbol{\psi}}^T \mathbf{J}_{\boldsymbol{\theta}})^T (\mathbf{J}_{\boldsymbol{\psi}}^T \mathbf{J}_{\boldsymbol{\psi}})^{-1} & \mathbf{I}_{\boldsymbol{\theta}} \end{pmatrix} \quad (16)$$

where $\mathbf{I}_{\boldsymbol{\psi}} \in \mathbb{R}^{L \times L}$ and $\mathbf{I}_{\boldsymbol{\theta}} \in \mathbb{R}^{K \times K}$ are identity matrices. Left-multiplying the left-hand side of Equation 15 by \mathbf{C} yields

$$\begin{pmatrix} \mathbf{J}_{\boldsymbol{\psi}}^T \mathbf{J}_{\boldsymbol{\psi}} & \mathbf{J}_{\boldsymbol{\psi}}^T \mathbf{J}_{\boldsymbol{\theta}} \\ \mathbf{0} & \mathbf{J}_{\boldsymbol{\theta}}^T \mathbf{J}_{\boldsymbol{\theta}} - (\mathbf{J}_{\boldsymbol{\psi}}^T \mathbf{J}_{\boldsymbol{\theta}})^T (\mathbf{J}_{\boldsymbol{\psi}}^T \mathbf{J}_{\boldsymbol{\psi}})^{-1} \mathbf{J}_{\boldsymbol{\psi}}^T \mathbf{J}_{\boldsymbol{\theta}} \end{pmatrix} \quad (17)$$

and the right-hand side gives

$$\begin{pmatrix} \mathbf{J}_\psi^T \Delta \mathbf{e} \\ \mathbf{J}_\theta^T \Delta \mathbf{e} - (\mathbf{J}_\psi^T \mathbf{J}_\theta)^T (\mathbf{J}_\psi^T \mathbf{J}_\psi)^{-1} \mathbf{J}_\psi^T \Delta \mathbf{e} \end{pmatrix} \quad (18)$$

It follows from Equation 17 that we can now solve the equations independently for $\Delta \theta$. The solution of the reduced system will be the same as the solution to the original system of equations.

Up to this point, this is a standard application of the Schur complement. However, when $\mathbf{J}_\psi^T \mathbf{J}_\psi$ is large, sparse, and potentially rank deficient, we need to take care when computing the matrix expression $(\mathbf{J}_\psi^T \mathbf{J}_\theta)^T (\mathbf{J}_\psi^T \mathbf{J}_\psi)^{-1} \mathbf{J}_\psi^T$ in the lower parts of Equation 17 and Equation 18. To this end, we first apply an RRQR decomposition to \mathbf{J}_ψ , which leads to

$$\mathbf{J}_\psi = \mathbf{Q} \mathbf{R} \mathbf{\Pi}^T \quad (19)$$

where $\mathbf{\Pi} \in \mathbb{R}^{L \times L}$ is a permutation matrix whose goal is to move the most linearly independent columns of \mathbf{J}_ψ to the front, $\mathbf{R} \in \mathbb{R}^{M \times L}$, and $\mathbf{Q} \in \mathbb{R}^{M \times M}$ an orthogonal matrix. We use Equation 19 to simplify the matrix expression as follows

$$\begin{aligned} & (\mathbf{J}_\psi^T \mathbf{J}_\theta)^T (\mathbf{J}_\psi^T \mathbf{J}_\psi)^{-1} \mathbf{J}_\psi^T \\ &= \mathbf{J}_\theta^T \mathbf{Q} \mathbf{R} \mathbf{\Pi}^T (\mathbf{\Pi} \mathbf{R}^T \mathbf{Q}^T \mathbf{Q} \mathbf{R} \mathbf{\Pi}^T)^{-1} \mathbf{\Pi} \mathbf{R}^T \mathbf{Q}^T \\ &= \mathbf{J}_\theta^T \mathbf{Q} \mathbf{R} \mathbf{\Pi}^T (\mathbf{\Pi} \mathbf{R}^T \mathbf{R} \mathbf{\Pi}^T)^{-1} \mathbf{\Pi} \mathbf{R}^T \mathbf{Q}^T \\ &= \mathbf{J}_\theta^T \mathbf{Q}_1 \mathbf{R}_1 \mathbf{\Pi}^T (\mathbf{\Pi} \mathbf{R}_1^T \mathbf{R}_1 \mathbf{\Pi}^T)^{-1} \mathbf{\Pi} \mathbf{R}_1^T \mathbf{Q}_1^T \\ &= \mathbf{J}_\theta^T \mathbf{Q}_1 \mathbf{R}_1 \mathbf{\Pi}^T \mathbf{\Pi} \mathbf{R}_1^{-1} \mathbf{R}_1^{-T} \mathbf{\Pi}^T \mathbf{\Pi} \mathbf{R}_1^T \mathbf{Q}_1^T \\ &= \mathbf{J}_\theta^T \mathbf{Q}_1 \mathbf{R}_1 \mathbf{R}_1^{-1} \mathbf{R}_1^{-T} \mathbf{R}_1^T \mathbf{Q}_1^T \\ &= \mathbf{J}_\theta^T \mathbf{Q}_1 \mathbf{Q}_1^T \end{aligned} \quad (20)$$

where $\mathbf{R}_1 \in \mathbb{R}^{L \times L}$ is the upper-triangular part of \mathbf{R} and $\mathbf{Q}_1 \in \mathbb{R}^{M \times L}$ contains the first L Householder vectors of \mathbf{Q} . In this derivation, we have first used the orthogonality of \mathbf{Q} to eliminate it from the matrix inversion. In a second step, we have converted the decomposition into a thin QR decomposition. Lastly, the orthogonality of $\mathbf{\Pi}$ has allowed us to arrive at the final result. Consequently, we have transformed an involved expression containing a large matrix inversion into a much affordable matrix multiplication. Furthermore, we are not affected by the numerical rank of \mathbf{J}_ψ since we do not compute any matrix inverse, and we only need a reduced QR decomposition.

Using Equation 20, we introduce the following auxiliary variables,

$$\mathbf{A}_\theta = \mathbf{J}_\theta^T \mathbf{J}_\theta - (\mathbf{J}_\theta^T \mathbf{Q}_1)(\mathbf{J}_\theta^T \mathbf{Q}_1)^T \quad (21)$$

$$\mathbf{b}_\theta = \mathbf{J}_\theta^T \Delta \mathbf{e} - (\mathbf{J}_\theta^T \mathbf{Q}_1)(\Delta \mathbf{e}^T \mathbf{Q}_1)^T \quad (22)$$

such that the subsystem of equations for $\Delta \theta$ becomes

$$\mathbf{A}_\theta \Delta \theta = \mathbf{b}_\theta \quad (23)$$

where $\mathbf{A}_\theta \in \mathbb{R}^{K \times K}$ and $\mathbf{b}_\theta \in \mathbb{R}^K$. When using the Householder form of \mathbf{Q}_1 , \mathbf{A}_θ and \mathbf{b}_θ can be computed at low cost. Moreover, even though \mathbf{A}_θ is dense, the small size of K keeps the memory footprint low. Prior to solving these equations, we factorize \mathbf{A}_θ using SVD such that

$$\mathbf{A}_\theta = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (24)$$

where $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_K)$ and $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_K)$ are orthogonal matrices, and $\mathbf{S} = \text{diag}[\mathbf{s}_1, \dots, \mathbf{s}_K]$ is a diagonal matrix of singular values.

We compute the numerical rank r_{ϵ_θ} of \mathbf{A}_θ from its singular values using a threshold ϵ_θ . This threshold is proportional to the magnitude of the expected noise in \mathbf{A}_θ (Hansen, 1988) and we shall further investigate its determination later in this paper. From the SVD and the numerical rank, we can readily solve Equation 23 using a truncated SVD (TSVD) (Hansen, 1987) into an observable and an unobservable part

$$\Delta \theta = \sum_{i=1}^{r_{\epsilon_\theta}} \frac{\mathbf{u}_i^T \mathbf{b}_\theta}{\mathbf{s}_i} \mathbf{v}_i \quad (25)$$

The computation of the full covariance matrix involves the inverse of the Fisher information matrix. However, for a 2×2 block matrix as in the left-hand side of Equation 15, we can advantageously perform blockwise inversions. Due to the marginalization property of the normal distribution, we can decompose the marginal covariance matrix as

$$\Sigma_\theta = \sum_{i=1}^{r_{\epsilon_\theta}} \frac{\mathbf{v}_i \mathbf{u}_i^T}{\mathbf{s}_i} + \sum_{i=r_{\epsilon_\theta}+1}^K \frac{\mathbf{v}_i \mathbf{u}_i^T}{\mathbf{s}_i} \quad (26)$$

Finally, we have a direct access to the numerical nullspace of \mathbf{A}_θ as

$$\text{null}[\mathbf{A}_\theta, \epsilon_\theta] = \text{span}\{\{\mathbf{v}_{r_{\epsilon_\theta}+1}, \dots, \mathbf{v}_K\}\} \quad (27)$$

Jauffret (2007) clearly articulates the link between the singularity of the Fisher information matrix and observability of the parameters being estimated. In our context, the basis vectors from the numerical nullspace correspond to directions in the parameter space which are numerically unobservable. While the first r_{ϵ_θ} column vectors of \mathbf{V} form an orthonormal basis for the row space of \mathbf{A}_θ or the observable subspace, the basis vectors from the nullspace form an orthonormal basis for the unobservable subspace.

Stacking the basis vectors of the nullspace into a matrix $\mathbf{N}_{\mathbf{A}_\theta} \in \mathbb{R}^{K \times (K-r_{\epsilon_\theta})}$, we reveal the local observability structure of the parameter. Each row i of the nullspace matrix represents the observability of the dimension i of the parameter vector. A row of zeros implies that the corresponding dimension is fully observable. A row containing a single one signifies that the corresponding dimension is unobservable. Any other rows correspond to dimensions in the parameter space that are only observable as a linear combination of other dimensions. We can interpret the TSVD solution in

Equation 25 as the unique least-squares solution in the observable subspace. In other words, the shift vector will only update the observable part of the parameter vector.

3.3.2. Nuisance variable estimation. Plugging $\Delta\theta$ into Equation 15, $\Delta\psi$ becomes the solution to the following system of equations

$$(\mathbf{J}_\psi^T \mathbf{J}_\psi) \Delta\psi = \mathbf{J}_\psi^T (\Delta\mathbf{e} - \mathbf{J}_\theta \Delta\theta) \quad (28)$$

which can be recognized as the normal equations of the least-squares problem

$$\min_{\Delta\psi \in \mathbb{R}^L} \|\mathbf{J}_\psi \Delta\psi - (\Delta\mathbf{e} - \mathbf{J}_\theta \Delta\theta)\|_2 \quad (29)$$

Given the potential large size of \mathbf{J}_ψ , we solve Equation 29 with an RRQR decomposition. In most applications, this is indeed the only possibility. Furthermore, this decomposition is already available from Equation 19 and we directly compute a basic solution as

$$\Delta\psi = \Pi \begin{pmatrix} \mathbf{R}_{11}^{-1} \mathbf{Q}_{11}^T (\Delta\mathbf{e} - \mathbf{J}_\theta \Delta\theta) \\ \mathbf{0} \end{pmatrix} \quad (30)$$

where $\mathbf{Q}_{11} \in \mathbb{R}^{M \times r_{\epsilon_\psi}}$ contains the first r_{ϵ_ψ} Householder vectors of \mathbf{Q} , $\mathbf{R}_{11} \in \mathbb{R}^{r_{\epsilon_\psi} \times r_{\epsilon_\psi}}$ is the upper-left triangular part of \mathbf{R} , and r_{ϵ_ψ} the estimated numerical rank of \mathbf{J}_ψ with tolerance ϵ_ψ . Instead of inverting \mathbf{R}_{11} , we solve the system by back substitution. The basic solution has at most r_{ϵ_ψ} non-zero components. If \mathbf{J}_ψ has full rank, the lower null vector disappears and Equation 30 yields the unique least-squares solution to Equation 29. In this paper, we have chosen the basic solution for its computational simplicity. The least-squares solution would have required a complete orthogonal decomposition. As our primary interest lies in estimating θ , we claim that the basic solution is suitable for ψ .

3.3.3. Improper posterior inference. Using the shift vectors computed in the previous sections, we can iterate to convergence without taking steps in the unobservable directions in parameter space. After convergence, we would like to recover the posterior marginal density. If the calibration parameter is partially observable or unobservable, the posterior marginal density in Equation 4 is improper. The covariance matrix, Σ_θ , is singular and the posterior is a degenerate normal distribution. However, we can still define proper density functions in the observable and unobservable subspaces. To this end, we shall first apply the following affine transformation to Θ

$$\tilde{\Theta} = \mathbf{V}^T \Theta \quad (31)$$

where \mathbf{V} is the orthogonal matrix from the SVD in Equation 24. Due to the affine transformation property of the normal distribution, the random vector $\tilde{\Theta}$ is normally distributed as

$$\tilde{\Theta} | \mathbf{x}_{1:N} \sim \mathcal{N}(\mathbf{V}^T \mu_\theta, \mathbf{V}^T \Sigma_\theta \mathbf{V}) \quad (32)$$

We can further partition $\tilde{\Theta}$ into an observable and unobservable part $\tilde{\Theta} = (\tilde{\Theta}_{obs}, \tilde{\Theta}_{nobs})^T$, where

$$\begin{aligned} \tilde{\Theta}_{obs} &= (\tilde{\Theta}_1, \dots, \tilde{\Theta}_{r_{\epsilon_\theta}})^T, \\ \tilde{\Theta}_{nobs} &= (\tilde{\Theta}_{r_{\epsilon_\theta}+1}, \dots, \tilde{\Theta}_K)^T \end{aligned} \quad (33)$$

We adopt the same partition for the mean vector of $\tilde{\Theta}$, i.e.

$$\mu_{\tilde{\Theta}} = \mathbf{V}^T \mu_\theta = (\mu_{\tilde{\Theta}_{obs}}, \mu_{\tilde{\Theta}_{nobs}})^T \quad (34)$$

Finally, we can use the orthogonality of \mathbf{V} to simplify the covariance matrix of $\tilde{\Theta}$ into

$$\Sigma_{\tilde{\Theta}} = \mathbf{V}^T \Sigma_\theta \mathbf{V} = \begin{bmatrix} \mathbf{S}_{\tilde{\Theta}_{obs}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{\tilde{\Theta}_{nobs}}^{-1} \end{bmatrix} \quad (35)$$

where \mathbf{S} is the diagonal matrix of the singular values of \mathbf{A}_θ defined in Equation 24

$$\mathbf{S} = \text{diag}[s_1, \dots, s_K] \quad (36)$$

and $\mathbf{S}_{\tilde{\Theta}_{obs}}$ and $\mathbf{S}_{\tilde{\Theta}_{nobs}}$ are, respectively, the upper-left and lower-right blocks of \mathbf{S} . Using the marginalization property of the normal distribution, the observable part is normally distributed as

$$\tilde{\Theta}_{obs} | \mathbf{x}_{1:N} \sim \mathcal{N}(\mu_{\tilde{\Theta}_{obs}}, \mathbf{S}_{\tilde{\Theta}_{obs}}^{-1}) \quad (37)$$

and the unobservable part is distributed as

$$\tilde{\Theta}_{nobs} | \mathbf{x}_{1:N} \sim \mathcal{N}(\mu_{\tilde{\Theta}_{nobs}}, \mathbf{S}_{\tilde{\Theta}_{nobs}}^{-1}) \quad (38)$$

Note that, since $\mathbf{S}_{\tilde{\Theta}_{nobs}}$ is singular, its inverse is undefined. This can be interpreted as a flat distribution, i.e. a normal with infinitely large covariance. Intuitively, since the data contain no information about $\tilde{\Theta}_{nobs}$, the posterior reverts to the (uninformative) prior.

3.4. Online inference

The method presented thus far is inherently offline, i.e. it requires an entire calibration dataset for inferring parameter values. For nonlinear least-squares problems, full batch methods are more accurate and stable as they iteratively refine the linearization point as compared to recursive estimation techniques such as the extended Kalman filter (EKF). In this paper, we want to benefit from the robustness of offline methods, along with the possibility to deploy our algorithm in an online and long-term setting. To reach this goal, we process small batches of data sequentially and decide to keep them based on their utility for the calibration. Each new batch is merged to the old ones to refine our knowledge about the calibration parameters until we reach a satisfactory level of confidence.

To start with the mathematical formulation of our online method, we define the current set of stored data samples as $\mathcal{D}^{info} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, which has led to the posterior marginal density in Equation 37, associated with the random variable $\tilde{\Theta}_{obs} | \mathcal{D}^{info}$. The set, \mathcal{D}^{info} , contains the current informative measurements for the calibration variable. Our sensor, \mathcal{S} , continuously streams new data that we accumulate in another batch of size ΔN denoted by $\mathcal{D}^{new} = \{\mathbf{x}_{N+1}, \dots, \mathbf{x}_{N+\Delta N}\}$. Intuitively, if the measurements in \mathcal{D}^{new} are similar to those in \mathcal{D}^{info} , we are not really improving our knowledge about $\tilde{\Theta}_{obs}$ and we can safely discard \mathcal{D}^{new} to keep the computation tractable.

Information theory provides a principled way to evaluate the usefulness of \mathcal{D}^{new} . Shortly after its introduction in the signal processing community, information theory has been adopted by statisticians. Lindley (1956) defined the amount of information in a random variable as the negative Shannon entropy. A random variable with high entropy, e.g. uniformly distributed, contains little information. Conversely, a sharply-peaked distribution with low entropy conveys high information. The amount of information provided by an experiment may therefore be defined as the difference between the posterior and prior information. This can be interpreted as a measure of the utility or information gain of the experiment. In our setup, we can consider the prior to be $\tilde{\Theta}_{obs} | \mathcal{D}^{info}$, the posterior $\tilde{\Theta}_{obs} | \mathcal{D}^{info}, \mathcal{D}^{new}$, and define the utility to be

$$\begin{aligned} \mathcal{U}(\tilde{\Theta}_{obs} | \mathcal{D}^{info}, \mathcal{D}^{new}) \\ = h(\tilde{\Theta}_{obs} | \mathcal{D}^{info}) - h(\tilde{\Theta}_{obs} | \mathcal{D}^{info}, \mathcal{D}^{new}) \end{aligned} \quad (39)$$

where $h(\cdot)$ denotes the differential entropy. It follows from Equation 39 that if \mathcal{D}^{new} is useful for estimating $\tilde{\Theta}_{obs}$, it reduces its entropy and increases its information. In other words, the utility tells us whether it is worth adding \mathcal{D}^{new} to our set of informative measurements. If the utility falls under a prescribed threshold $\delta_{\mathcal{U}}$, \mathcal{D}^{new} can be discarded since it does not bring a significant information gain. Otherwise, it can be added to the informative set which becomes $\mathcal{D}^{info} \cup \mathcal{D}^{new}$.

We shall now examine the derivation of the utility function in our context where the random variables are normally distributed. The differential entropy of a normally distributed random variable $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}}, \boldsymbol{\Sigma}_{\mathbf{X}})$ of dimension K is given by

$$h(\mathbf{X}) = \frac{K}{2} \left(1 + \log_e(2\pi) \right) + \frac{1}{2} \log_e |\boldsymbol{\Sigma}_{\mathbf{X}}| \quad (40)$$

where $|\cdot|$ denotes the matrix determinant and \log_e the natural logarithm. This latter is chosen for mathematical convenience, given the exponential form of the normal distribution. We can retrieve an entropy in bits by dividing Equation 40 by $\log_e(2)$. If we now state the distribution of

the prior as $\tilde{\Theta}_{obs} | \mathcal{D}^{info} \sim \mathcal{N}(\boldsymbol{\mu}^{prior}, \boldsymbol{\Sigma}^{prior})$ and of the posterior as $\tilde{\Theta}_{obs} | \mathcal{D}^{info}, \mathcal{D}^{new} \sim \mathcal{N}(\boldsymbol{\mu}^{post}, \boldsymbol{\Sigma}^{post})$, the information gain in Equation 39 becomes

$$\mathcal{U}(\tilde{\Theta}_{obs} | \mathcal{D}^{info}, \mathcal{D}^{new}) = \frac{1}{2} \log_e \frac{|\boldsymbol{\Sigma}^{prior}|}{|\boldsymbol{\Sigma}^{post}|} \quad (41)$$

As $\boldsymbol{\Sigma}^{post}$ and $\boldsymbol{\Sigma}^{prior}$ are diagonal matrices, their determinant is the product of the respective inverse singular values. Finally, when using the property of the logarithm, Equation 41 becomes

$$\mathcal{U}(\tilde{\Theta}_{obs} | \mathcal{D}^{info}, \mathcal{D}^{new}) = \frac{1}{2} \left(\sum_{i=1}^{r_{\epsilon_{\theta}}^{post}} \log_e s_i^{post} - \sum_{i=1}^{r_{\epsilon_{\theta}}^{prior}} \log_e s_i^{prior} \right) \quad (42)$$

where $r_{\epsilon_{\theta}}^{post}$ and $r_{\epsilon_{\theta}}^{prior}$ are the numerical ranks of $\mathbf{A}_{\theta}^{post}$ and $\mathbf{A}_{\theta}^{prior}$ respectively. In summary, the information gain in Equation 42 is available at low cost from the already computed SVD. We can simply maintain the sum of the logarithm of the singular values of the current estimate. When $r_{\epsilon_{\theta}}^{post} > r_{\epsilon_{\theta}}^{prior}$, the data has increased the dimensionality of the observable subspace. This corresponds to infinite information gain and the batch is always accepted.

3.5. Algorithmic details

For a given sensor, \mathcal{S} , we require a reasonable initial guess of the calibration ($\theta_{(0)}$) and nuisance ($\psi_{(0)}$) parameters, the forward model, $g_{\mathcal{S}}(\cdot)$, describing the data generation process, the analytical or numerical Jacobian matrix, \mathbf{J} , of the error terms, and the data covariance matrix, $\boldsymbol{\Sigma}_{\mathbf{X}}$. This latter is either available from the datasheet of the sensor or should be estimated from a training dataset.

In addition to the model parameters, our method relies on algorithmic parameters that can be classified into the sets $\mathcal{P}_N = \{\epsilon_{\psi}, \epsilon_{\theta}\}$ and $\mathcal{P}_O = \{\delta_{\mathcal{U}}, \Delta N\}$. These parameters are specific to our algorithm and govern its numerical and online behavior, respectively. In \mathcal{P}_N , the parameters ϵ_{ψ} and ϵ_{θ} are the thresholds for the numerical rank detection of \mathbf{J}_{ψ} and \mathbf{A}_{θ} defined above. Their value depends on the expected noise in the matrix entries, which is composed of roundoff errors and, in our case, essentially of discrepancies between the current estimate and the true value for the parameters. Except when we have a statistical model for the noise as Hansen (1988), the determination of these parameters remains empirical and application specific. In \mathcal{P}_O , the parameter ΔN controls the size of the batches evaluated and $\delta_{\mathcal{U}}$ is a threshold for the information provided by a batch with respect to an existing marginal posterior for $\tilde{\Theta}$. Here ΔN can represent a certain number of measurements or a time window during which measurements are gathered. A large value for ΔN corresponds in the limit to a full batch method and is therefore not recommended for online operations. A

small value for ΔN involves many unnecessary optimization runs and might not provide enough information to render the calibration parameters observable. The selection of ΔN is thus purely empirical, dependent on the vehicle dynamics, and is a trade-off between online and offline behavior. The information threshold δ_U is given in bits. If it is set to a low value, most of the batches will be accepted and our algorithm will become largely inefficient as compared to a full batch method. On the other hand, when setting δ_U to a large value, we will discard most of the batches and fail to capture important information. As for ΔN , the choice of δ_U is empirical and application specific. These two parameters are furthermore related in the sense that larger batches can potentially reduce the uncertainty of the posterior distribution in a more significant way.

As mentioned earlier, robotic applications involving landmark observations or robot's poses will most likely result in a sparse Jacobian matrix. This specific model structure can be exploited by storing it in memory using a sparse matrix data structure, for which there exists an efficient QR decomposition (Davis, 2011). Consequently, our algorithm has a time complexity in $O(L^{3/2})$ (George and Ng, 1988) and space complexity in $O(L)$.

4. Experiments

In order to evaluate and validate the approach proposed in this paper, we have conducted experiments on simulated and real-world data. We start with a standard problem, namely a differential-drive wheeled robot observing landmarks with a LRF. The low dimensionality of this three-degree-of-freedom estimation problem eases the thorough analysis of the multiple claims outlined in Section 3 and provides a proof of concept. In a second experiment, we tackle a high-dimensional calibration problem involving a visual-inertial sensor unit. As the unit undergoes different motion patterns, different extrinsic calibration parameters become observable. We show how the degree of observability of these parameters is inferred from the data automatically and incrementally within our calibration framework.

4.1. LRF on a differential-drive robot

We consider a differential-drive wheeled robot equipped with a LRF. While moving in a plane, the LRF measures distances to poles disseminated on the ground. Furthermore, the robot is endowed with odometry sensors, measuring its linear and angular velocity. This scenario is a prototypical SLAM problem. Freely adopting the SLAM jargon, we refer to the poles as landmarks. The positions in an inertial reference frame of a collection of landmarks form a map of the environment. Given an initial inertial reference frame, the SLAM problem consists in estimating the poses of the robot, i.e. its trajectory, and the positions of the landmarks, solely based on its sensors. In order to solve this problem, we require the relative pose between the LRF frame and the robot frame, and the relative poses

between the robot and the inertial frame. The LRF measurements shall indeed be transformed in the inertial frame to construct a consistent map. Recalling our presentation in Section 3, the relative pose between the LRF frame and the robot frame corresponds to the calibration variable, and the relative poses between the robot and the inertial frame, along with the positions of the landmarks, to the nuisance variable.

This scenario was first showcased in Maye et al. (2013) with a preliminary version of our algorithm. In the rest of this section, we shall phrase the underlying mathematical model and explore the problem to its full extent. Due to its simplicity, we can thoroughly analyze the behavior of our algorithm with extensive simulations and real-world experiments.

4.1.1. Problem formulation. Before starting, we shall first identify the different coordinate frames in the problem. The inertial reference frame or world frame is represented by \mathcal{F}_W with origin w . As we only compute relative poses to \mathcal{F}_W , its absolute pose is arbitrary. The robot frame, \mathcal{F}_R , has its origin, r , located at the center of the axle between the left and right wheels. The first axis of \mathcal{F}_R is aligned with the direction of forward motion of the robot and the second axis points to the left. The LRF frame, \mathcal{F}_L , has the origin, l , at the point of emission of the laser beams. The first axis of \mathcal{F}_L points along the forward direction of the LRF and the second axis to the left.

The relative pose of \mathcal{F}_R with respect to \mathcal{F}_W is a time-varying transformation denoted by ${}^w\mathbf{T}_r(t)$, composed of a rotation, ${}^w\mathbf{R}_r(t)$, and a translation, ${}^w\mathbf{r}^r(t)$. The relative pose of \mathcal{F}_L with respect to \mathcal{F}_R is a time-invariant transformation denoted by ${}^r\mathbf{T}_l$, composed of a rotation, ${}^r\mathbf{R}_l$, and a translation, ${}^r\mathbf{r}^l$. The motion of the robot being constrained on a plane, we represent a translation, respectively a linear velocity, by two scalar parameters, and a rotation, respectively an angular velocity, by a single angular parameter.

Motion model For the purpose of this application, we choose to approximate the motion model in discrete time using a first-order Runge–Kutta method with a small step size, T , as

$$\begin{aligned} {}^w\mathbf{r}_{k+1}^{wr} &= {}^w\mathbf{r}_k^{wr} + T {}^w\mathbf{R}_k {}^r\mathbf{v}_k^{wr}, \\ \varphi_{k+1} &= \varphi_k + T \omega_k, \\ 0 &= (0 \quad 1) {}^r\mathbf{v}_k^{wr}, \end{aligned} \quad (43)$$

where k is the discrete-time index, ${}^r\mathbf{v}_k^{wr}$ the linear velocity of \mathcal{F}_R as seen from \mathcal{F}_W and expressed in \mathcal{F}_R , ω_k the angular velocity of \mathcal{F}_R as seen from \mathcal{F}_W , and

$${}^w\mathbf{R}_k \begin{pmatrix} \cos \varphi_k & -\sin \varphi_k \\ \sin \varphi_k & \cos \varphi_k \end{pmatrix} \quad (44)$$

The last equation in Equation 43 acts as a non-holonomic constraint disallowing pure lateral motion of the robot.

The robot can measure its linear and angular velocity through rotary encoders attached to its wheels. We adopt here a simplified odometry model that provides at time k a measurement, $\mathbf{v}_k^o = (v_k^o, 0, \omega_k^o)^T$, composed of a linear velocity, v_k^o , and an angular velocity, ω_k^o , in an odometry frame, \mathcal{F}_O , collocated with \mathcal{F}_R . By adding a pseudo zero measurement for satisfying the non-holonomic constraint, we can isolate ${}^r\mathbf{v}_k^{wr}$ and ω_k in Equation 43 to relate it to the odometry measurement.

Observation model The environment contains a set of Nl landmarks represented by the points $\{\ell_1, \dots, \ell_N\}$. We assume that the LRF returns for each landmark, ℓ_i , a range, r^{ℓ_i} , and a bearing angle, φ^{ℓ_i} . If a landmark is not in the field of view of the LRF, its corresponding range is set to zero and flagged as invalid. Despite being a rather simplified setup, it is perfectly sufficient for demonstrating our approach, while circumventing the problem of landmark detection and association between consecutive laser scans. The range r^{ℓ_i} corresponds to the distance between the origin, l , and the landmark, ℓ_i . The bearing angle, φ^{ℓ_i} , corresponds to the angle between the vector, ${}^l\mathbf{r}^{\ell_i}$, and the first axis of \mathcal{F}_L . The ranges and bearing angles can also be interpreted as the polar coordinates of the landmarks with respect to a polar coordinate system \mathcal{F}_L , whose pole is located at the point l and polar axis along the first axis of \mathcal{F}_L . Therefore, the measurement at time k of a landmark ℓ_i can be represented by the column vector

$${}^i\ell_{ik} = \begin{pmatrix} r_k^{\ell_i} \\ \varphi_k^{\ell_i} \end{pmatrix} = {}^\circ f_\perp({}^l\mathbf{T}_r {}^r\mathbf{T}_{wk} {}^w\ell_i) \quad (45)$$

where the function ${}^\circ f_\perp$ converts from Cartesian to polar coordinates and ${}^w\ell_i$ contains the Cartesian coordinates of ℓ_i with respect to \mathcal{F}_W .

Estimation model While Equation 43 provides a forward model $g_o({}^w\mathbf{T}_{r_{k+1}}, {}^w\mathbf{T}_{r_k})$ for an odometry measurement, \mathbf{v}_k^o , Equation 45 defines a forward model $g_{\ell_i}({}^w\mathbf{T}_{r_k}, {}^w\ell_i, {}^r\mathbf{T}_l)$ for an LRF measurement, ${}^i\ell_{ik}$. We denote the covariance matrix for \mathbf{v}_k^o by $\Sigma_o = \text{diag}[\sigma_v^2, \sigma_c^2, \sigma_\omega^2]$ and the covariance matrix for ${}^i\ell_{ik}$ by $\Sigma_\ell = \text{diag}[\sigma_r^2, \sigma_\varphi^2]$. For this application, we assume that we obtain measurements from each sensor synchronously, i.e. for each timestep, k , we observe \mathbf{v}_k^o and ${}^i\ell_{1:Nl_k}$.

With regard to Section 3, a data sample of size N consists in $\{\mathbf{v}_{1:N}^o, {}^i\ell_{1:Nl:N}\}$, the calibration variable in $\Theta = \{{}^r\mathbf{T}_l\}$, and the nuisance variable in $\Psi = \{{}^w\mathbf{T}_{r_{1:N}}, {}^w\ell_{1:Nl}\}$. Hence, the estimation problem can thus be formulated as the computation of the marginal posterior density $f({}^r\mathbf{T}_l | \mathbf{v}_{1:N}^o, {}^i\ell_{1:Nl:N})$, from the joint posterior density $f({}^r\mathbf{T}_l, {}^w\mathbf{T}_{r_{1:N}}, {}^w\ell_{1:Nl} | \mathbf{v}_{1:N}^o, {}^i\ell_{1:Nl:N})$.

We can obtain an initial estimate for ${}^w\mathbf{T}_{r_{2:N}}$ by successively applying the discretized motion model in Equation 43 with the odometry measurements, $\mathbf{v}_{1:N}^o$, starting from an

arbitrary location, ${}^w\mathbf{T}_{r_1}$. The initial guess for ${}^r\mathbf{T}_l$ comes from a measuring tape and an angle finder. For ${}^w\ell_{1:Nl}$, we can use the inverse model of Equation 45 using the initial guesses for ${}^w\mathbf{T}_{r_{1:N}}$ and ${}^r\mathbf{T}_l$.

4.1.2. Observability analysis. Before we delve into experimental evaluation, we shall perform an a priori analysis of the model. After an intuitive observability analysis, we shall validate our thoughts by the examination of the Jacobian matrix structure and the cost function graph. We note that this analysis is not necessary for our algorithm to work properly, but nevertheless provides valuable insights for interpreting the experimental results and for a proper understanding of the behavior of our method.

Forward model analysis We shall first start this analysis by considering a subset of the estimation problem, i.e. the estimation of robot's poses using odometry measurements. This simple example aims at illustrating the procedure that will drive the observability analysis of the full estimation problem.

Extending the discussions in Section 3, observability pertains to the injectivity of the forward model, i.e. whether it is possible to uniquely determine an inverse model. Assuming a sample of odometry data, $\mathbf{v}_{1:N}^o$, we inspect the joint forward model, $\mathbf{v}_{1:N}^o = G_o({}^w\mathbf{T}_{r_{1:N}})$, that generates it. In $G_o(\cdot)$, ${}^w\mathbf{T}_{r_{1:N}}$ forms the domain of the function. If distinct instances of ${}^w\mathbf{T}_{r_{1:N}}$ map to the same outputs $\mathbf{v}_{1:N}^o$, the parameters are not observable. In other words, the inverse function $G_o^{-1}(\cdot)$ is not uniquely determined by $G_o(\cdot)$.

For the sake of the example, if we withdraw the non-holonomic constraint on the lateral linear velocity of the robot, there are an infinite number of robot's poses, ${}^w\mathbf{T}_{r_{1:N}}$, that generate the same odometry outputs. For instance, the set of robot's poses with identical orientations along the second axis of the robot's frame will always be reflected by a constant zero output on the odometry sensors. Consequently, robot's poses are not observable. On the other hand, the zero velocity constraint on the second component of the linear velocity in the robot frame disallows these configurations and results in full observability. Our informal explanation is in line with the analysis conducted by Martinelli et al. (2006), which suggests a pose parameterization in polar coordinates to solve the problem.

Along this line of thoughts, we can proceed to the full joint forward model including the landmark positions and calibration parameters,

$$\begin{pmatrix} \mathbf{v}_{1:N}^o \\ {}^i\ell_{1:Nl:N} \end{pmatrix} = G({}^w\mathbf{T}_{r_{1:N}}, {}^w\ell_{1:Nl}, {}^r\mathbf{T}_l) \quad (46)$$

Because the model is under-constrained, there are an infinite number of elements of the domain of $G(\cdot)$ that map to the same element in its codomain. For instance, any affine transformation applied simultaneously to ${}^w\mathbf{T}_{r_{1:N}}$, ${}^w\ell_{1:Nl}$, and ${}^r\mathbf{T}_l$, gives rise to identical outputs. If the relative poses, ${}^w\mathbf{T}_{r_{1:N}}$, have the same orientation, an arbitrary translation

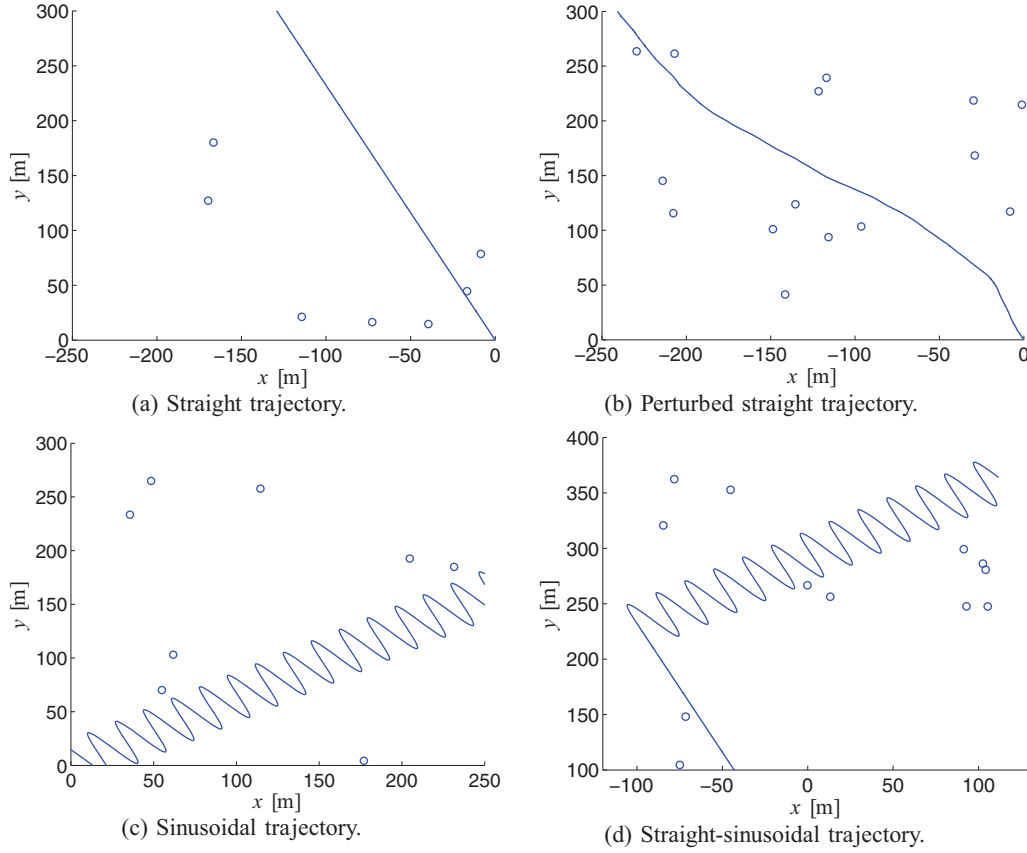


Fig. 1. Examples of robot trajectories used in our simulation experiments. The blue line corresponds to the robot's path, ${}^w\mathbf{T}_{r_{1:N}}$, and the blue circles to the landmark positions, ${}^w\ell_{1:N}$. In (a), only the rotational part of ${}^r\mathbf{T}_l$ is observable. In (b), corresponding to noisy odometry when the robot is actually driving straight, the translational part of ${}^r\mathbf{T}_l$ appears observable due to noise in the Jacobian matrix. (c) illustrates the fully observable case when the robot undergoes general motion, and (d) shows the combination of a partially and fully observable cases.

applied to ${}^r\mathbf{T}_l$ and to ${}^w\ell_{1:N}$ with respect to \mathcal{F}_W has no influence on the outputs. This situation corresponds to the robot driving in a straight line or standing still, with the LRF coordinate system translated along with the landmark positions. Although this rough analysis does not assert an exact figure, the identification of these unobservable directions can guide our experimental evaluation.

In the physical world, the inputs and outputs of $G(\cdot)$ are random quantities associated with a continuous probability density function. Although the probability for the realization of any of these degenerate cases is zero by definition, we might be arbitrarily close to one of them and our estimation problem will appear to be ill-conditioned. For this reason, it only makes sense to speak about numerical observability when a real physical system is involved.

Jacobian matrix structure Following on this informal analysis, we shall now examine if the structure of the Jacobian matrix supports our claims regarding the identified unobservable cases. As can be understood from the model in Section 4.1.1, no data appears in the Jacobian matrix. We can thus evaluate it for any realization of the latent variables given an unobserved data sample.

For this analysis, we concentrate on the cases depicted in Figure 1. The landmark positions, ${}^w\ell_{1:N}$, are sampled uniformly within a region around the robot's trajectory, ${}^w\mathbf{T}_{r_{1:N}}$, and the calibration variable, ${}^r\mathbf{T}_l$, is chosen arbitrarily. The trajectory is perfectly straight in Figure 1a, but in Figure 1b we slightly perturbed the straight path to resemble noisy odometry received from a robot driving in a straight line. From a linear algebra perspective, the evaluated Jacobian matrix from Figure 1b is a perturbed version of the one from Figure 1a. We shall finally focus on the situation sketched in Figure 1c where the robot's trajectory, ${}^w\mathbf{T}_{r_{1:N}}$, is modeled as a sinusoidal path and hence contains multiple changes of orientation.

By selecting a reasonable data sample size, we can analyze the structure of the full scaled Jacobian matrix, $[\mathbf{J}_\psi, \mathbf{J}_\theta]$, by SVD. The computed singular values for the three cases are shown in Figure 2a. For all cases, the three singular values clamped at zero correspond to the degrees of freedom (two for translation and one for orientation) induced by the lack of global anchor of the model. In the straight trajectory case, two singular values associated with the translation of ${}^r\mathbf{T}_l$ and ${}^w\ell_{1:N}$ remain at zero. In the perturbed straight trajectory, these two singular values depart

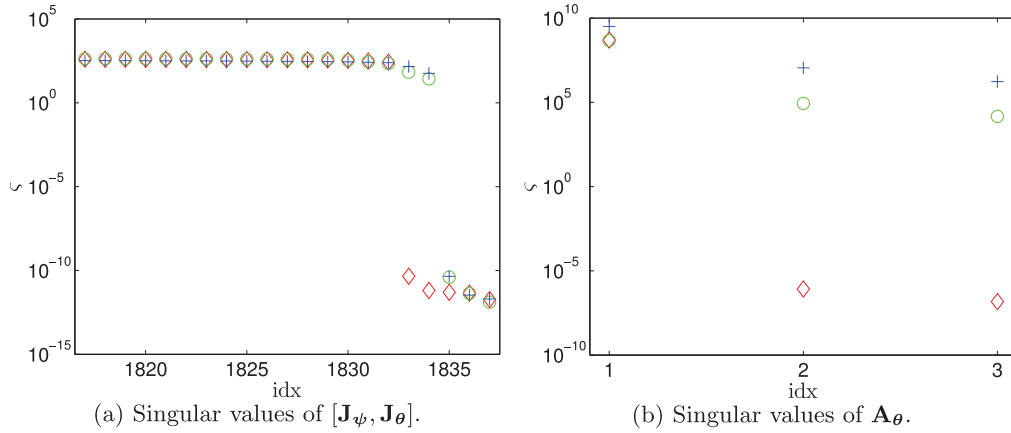


Fig. 2. The 20 smallest singular values of $[J_\psi, J_\theta]$ (Figure 2a) and of A_θ (Figure 2b) for the straight (red diamonds), perturbed straight (green circles), and sinusoidal (blue pluses) trajectory. In the perturbed straight scenario, the noisy matrix entries lead to two singular values clearly departing from zero, incorrectly rendering the translation parameters of the LRF observable.

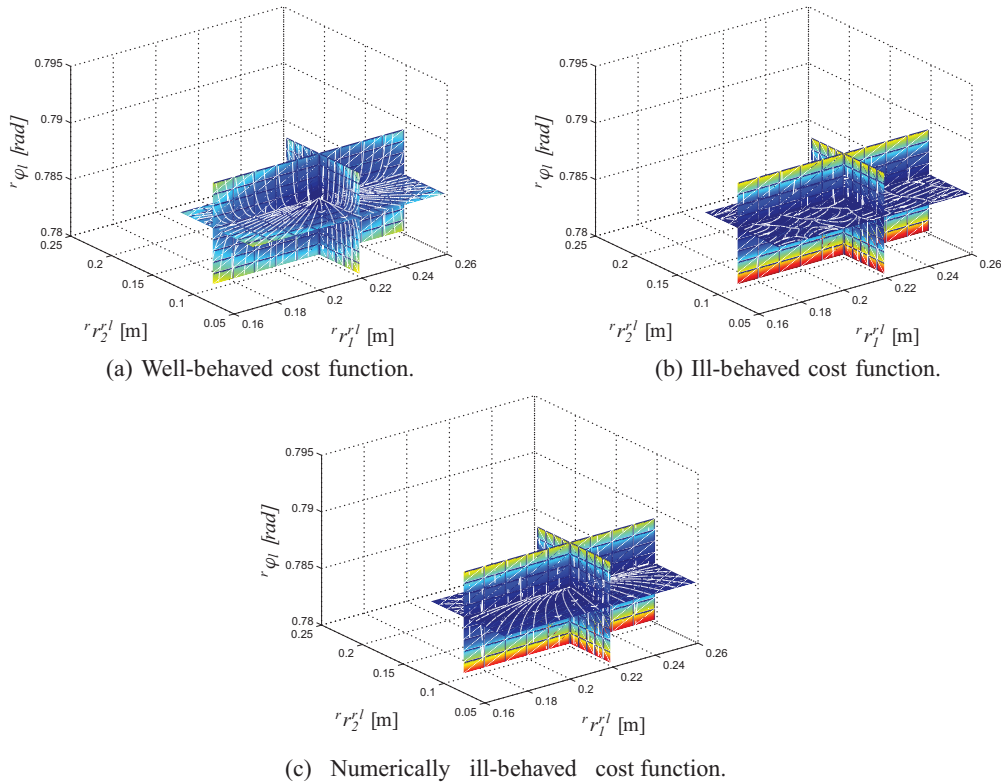


Fig. 3. Cost function for the fully observable (a), partially observable (b), and numerically unobservable (c) scenario. The planes converge at the ground truth value for the calibration parameters, low cost values are blue, higher cost values move through cyan, yellow, and red, and the gradient of the cost function is represented by the white arrows. Although (c) is a noisy version of (b), it incorrectly appears well-behaved.

from zero, which shall lead to an incorrect rank determination. The sinusoidal path being well-behaved, the singular values are legitimately above zero. After the SVD analysis of the full Jacobian matrix, we display in Figure 2b the singular values of the marginal Jacobian matrix, A_θ . This plot conveys essentially the same information and thus validates our intuition.

Cost function visualization To conclude this a priori analysis, it is worth picturing the cost function defined by Equation 6 as a function of the calibration parameters, ${}^w\mathbf{T}_l$. In Figure 3a, we have fixed the nuisance variables, ${}^w\mathbf{T}_{r_{1:N}}$ and ${}^w\ell_{1:N}$, corresponding to the sinusoidal path in Figure 1c and evaluated the cost function at different values of the calibration variable, ${}^r\mathbf{T}_l$. In this fully observable

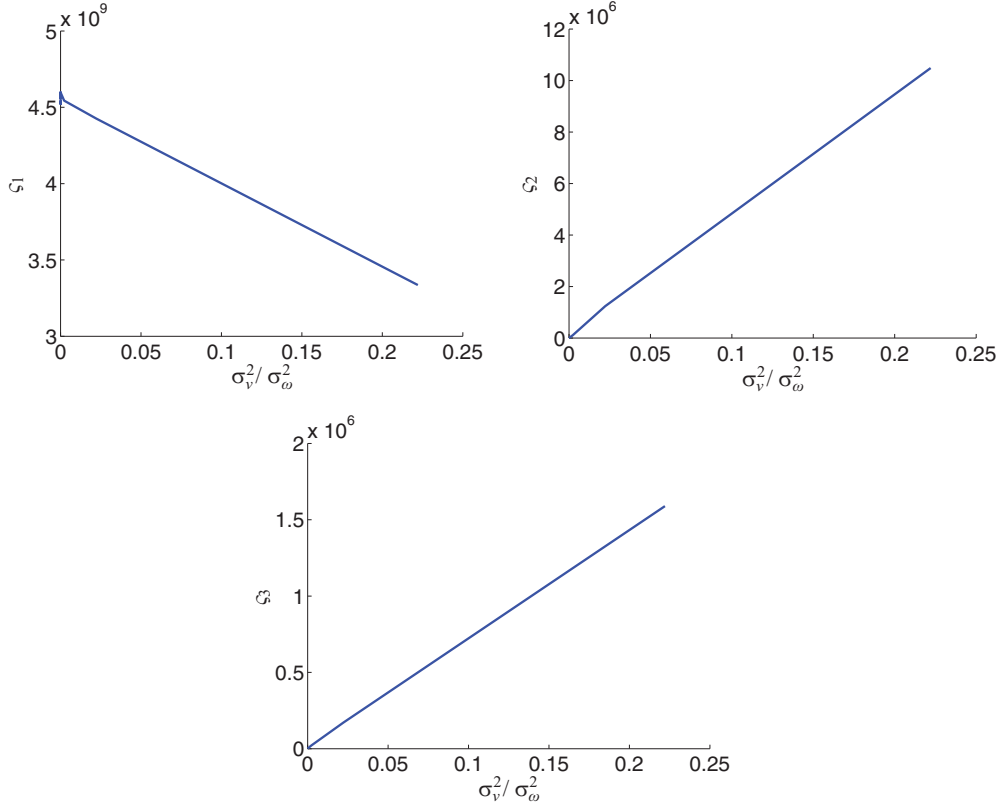


Fig. 4. Evolution of the singular values, $s_{1:3}$, of \mathbf{A}_θ as a function of odometry noise on a straight trajectory. In this scenario, s_2 and s_3 should be theoretically clamped at zero.

situation, we notice that the cost function is well-behaved, i.e. the gradient lines are convergent. To illustrate the behavior of the cost function in a partially observable setting, we display in Figure 3b the cost function corresponding to the straight trajectory in Figure 1a. In this plot, we have applied the same translation to ${}^r\mathbf{T}_l$ and ${}^w\ell_{1:N_b}$, resulting in identical costs for the translation parameters of ${}^r\mathbf{T}_l$. Consequently, the gradient lines in this plane are divergent and a nonlinear optimization algorithm without regularization will most likely fail. Finally, Figure 3c demonstrates the numerically unobservable situation depicted in Figure 1b. Although the input data contains just a slight deviation of the previous case, we witness here that the cost function appears well-behaved.

4.1.3. Simulation experiments. As we have an analytical formulation of the forward model for each sensor, we can select any configuration of the latent variables and generate measurements using their data distribution. For generating robot's poses, ${}^w\mathbf{T}_{r_{1:N}}$, we have chosen a sinusoid function, parameterized by an amplitude, A , and a frequency, f , providing at will straight to highly curved trajectories. This allows us to explore the sliding scale of numerical observability, i.e. the completely unobservable case when $A = 0$, weakly observable cases when A is small, and fully observable cases when A is large.

Numerical rank and observability detection In a first experiment, we report in Figure 4 the evolution of the singular values of the marginal Hessian matrix, \mathbf{A}_θ , as a function of the odometry noise in the straight trajectory example depicted in Figure 1a. For each noise level, we have averaged 10 simulations runs of 500 seconds. The odometry variances, σ_v^2 and σ_w^2 , range from 0 to 0.2. In the three plots, we observe that the singular values evolve linearly with the noise. For s_1 , the noise induces a slight decrease, which shall play no role in the numerical rank determination.

To conclude this part on the numerical rank and observability, we display in Figure 5 the evolution of the singular values of \mathbf{A}_θ as a function of the amplitude, A , of the sinusoidal trajectory for a fixed sample size $N = 5000$, $\sigma_v^2 = 10^{-4}$, and $\sigma_w^2 = 10^{-4}$. We have selected here a conservative numerical rank threshold, $\epsilon_\theta = 5 \times 10^4$, such that the calibration parameters become fully observable for an amplitude of approximatively $A \approx 0.45$ [m].

In all our experiments, the numerical rank, r_{ϵ_θ} , of \mathbf{J}_{ψ_l} (rank deficiency of 3 for the global transformation of all of the frames) was correctly detected by the RRQR decomposition without any particular attention to the threshold ϵ_ψ determined by the machine epsilon.

Qualitative evaluation We shall now demonstrate the overall behavior of our algorithm on some illustrative examples. It is for instance of interest to monitor the

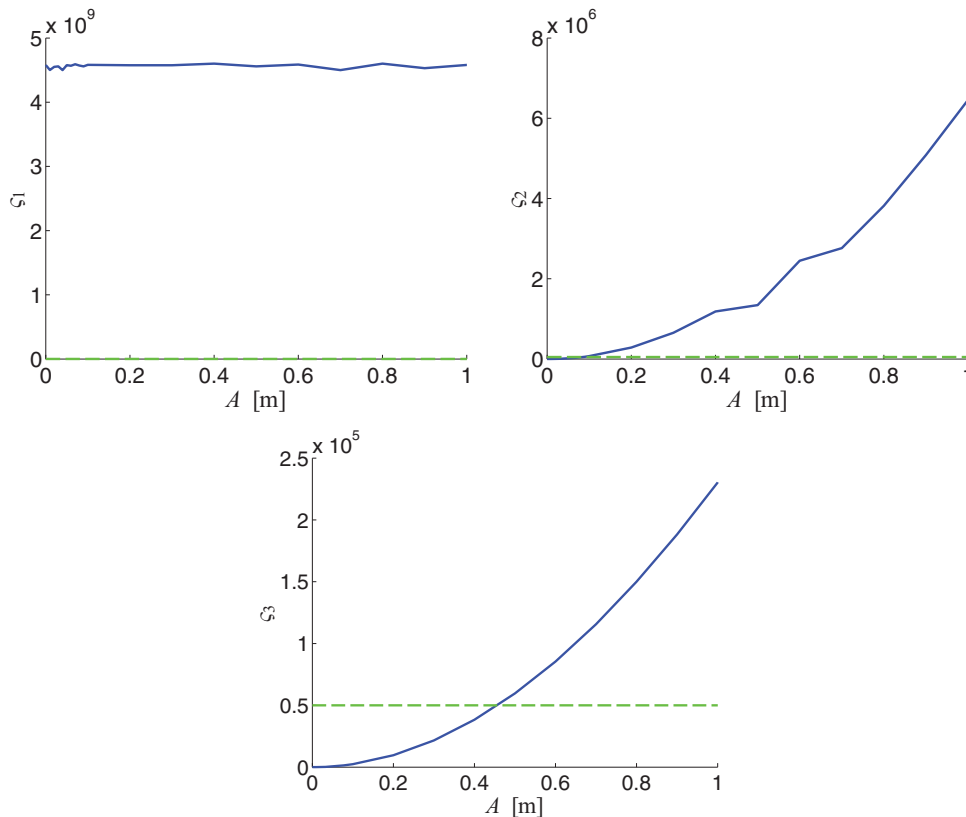


Fig. 5. Evolution of the singular values (blue line), $\varsigma_{1:3}$, of \mathbf{A}_θ as a function of the amplitude, A , of a sinusoidal path. With the selected numerical rank threshold (green line), $\epsilon_\theta = 5 \times 10^4$, the calibration parameters become fully observable at $A \approx 0.45$ [m].

measurement batches considered as informative and whether the selection is consistent with our intuition. We shall also visualize the evolution of the posterior distribution over the calibration parameters as new batches are added. We will stick to the three scenarios presented thus far, i.e. a straight trajectory (Figure 1a), a straight trajectory followed by a sinusoidal path (Figure 1d), and a sinusoidal path (Figure 1c).

In Figure 6, we summarize the relevant qualitative results of our simulation experiments. The first column corresponds to a straight (partially observable), the second to a sinusoidal (fully observable), and the third to the combination of a straight and sinusoidal trajectory. In the first row, we show the selected informative batches, along with the estimated robot's poses and landmark positions. Although these estimates are not of primary interest, their proximity to the ground truth is a pleasing indicator of the performance of our algorithm. For visualization, the estimates were aligned with the ground truth by aligning the landmarks (Fiore, 2001). Out of 25 batches, the straight trajectory used 4, the sinusoidal 11, and the straight-sinusoidal trajectory 15. In order to reveal the effect of batch selection on the calibration parameters, we display in the second to fourth rows, the evolution of the parameters as a function of the incoming batches. As expected, we witness that the parameters remain at their initial guess until they become

observable. Finally, we show in the last row the evolution of the information gain as new batches are added to the online estimator. The behavior of the entropy difference concurs with our intuition.

Quantitative evaluation To complete the simulation experiments, we compare the performance of our algorithm against a full-batch LM algorithm and an EKF. We generate a sinusoidal trajectory with varying amplitude such that it covers partial (low amplitude) to full observability (high amplitude) of the calibration parameters. To get significant statistical results, we repeat the experiment 100 times for a given amplitude and use the same initial conditions for the three methods at each run. While keeping identical calibration parameters over the entire experiment, we sample new landmark positions and orientation of the robot's trajectory at each run. The sinusoid function amplitudes range from $A = 0$ [m] to $A = 1$ [m], with steps of 0.01 [m] until $A = 0.1$ [m], and 0.1 [m] until $A = 1$ [m].

The estimated calibration parameters for all methods are displayed in Figure 7. As we have noticed that the EKF is highly sensitive to the concentration of the initial prior on the calibration parameters, we have included three versions with different priors for comparison. While a tight prior induces little changes on the initial guess or a bias, a loose prior yields instable estimates with many outliers. The best-performing EKF has required a tedious fine-tuning.

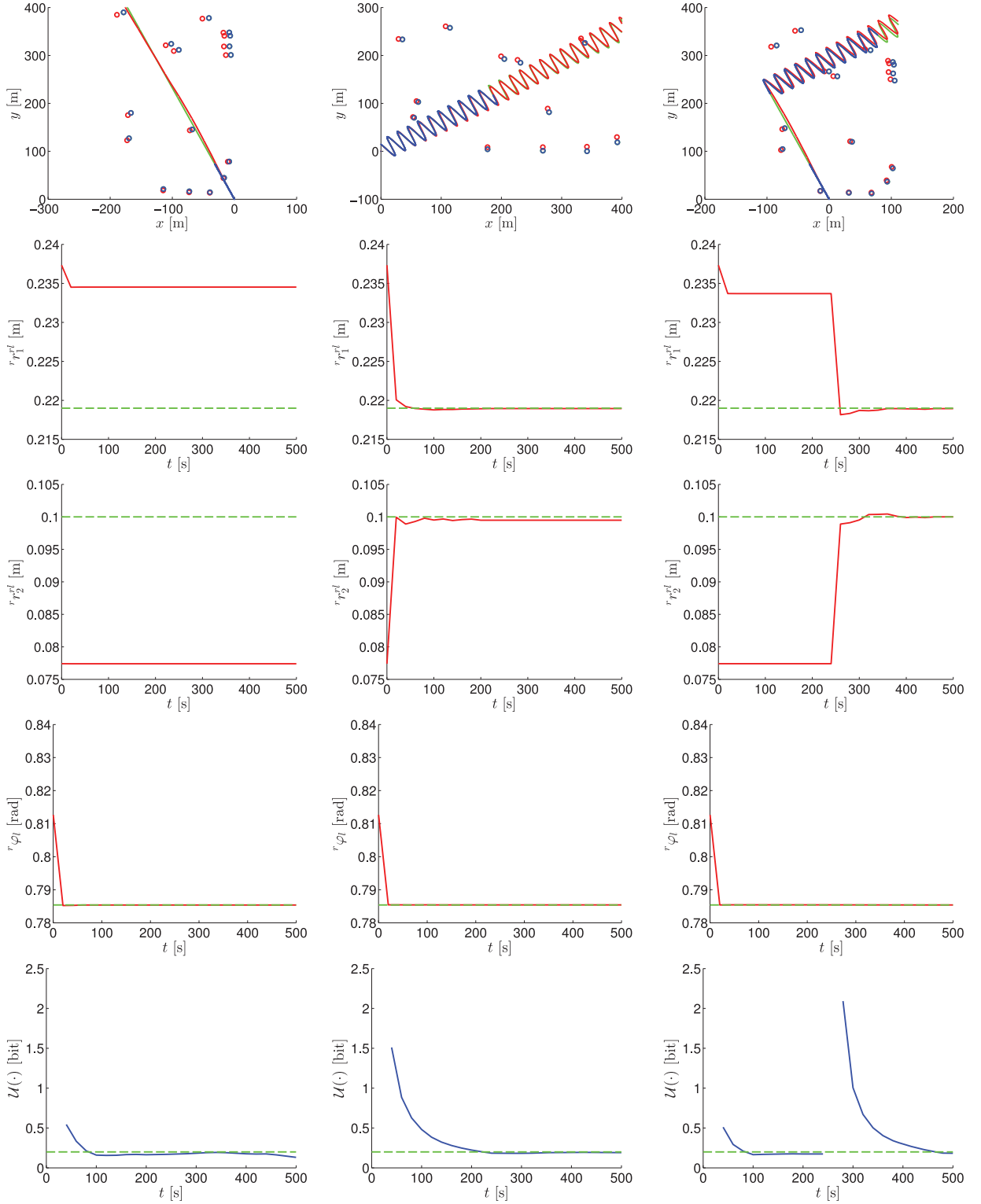


Fig. 6. Qualitative results for the simulation experiments. In the first row, the ground truth trajectory, ${}^w\mathbf{T}_{r_{1:N}}$, is shown in green with initial guess in red, ground truth landmark positions, ${}^w\ell_{1:N}$, in green circles with initial guess in red circles, trajectory estimate, ${}^w\hat{\mathbf{T}}_{r_{1:N}}$, in blue, and landmark positions estimate, ${}^w\hat{\ell}_{1:N}$, in blue circles. The second, third, and fourth rows show the calibration parameter estimates, ${}^r\hat{\mathbf{T}}_l = ({}^r\hat{r}_1^d, {}^r\hat{r}_1^d, {}^r\hat{\varphi}_l)$ (red), as a function of the incoming batches, with the ground truth, ${}^r\mathbf{T}_l$, in green. The last row shows the information gain as a function of the incoming batches, with the information gain threshold, $\delta_{\mathcal{U}}$, in green. The first column is representative for a partially observable, the second for a fully observable, and the last for a combination of partially and fully observable scenario.

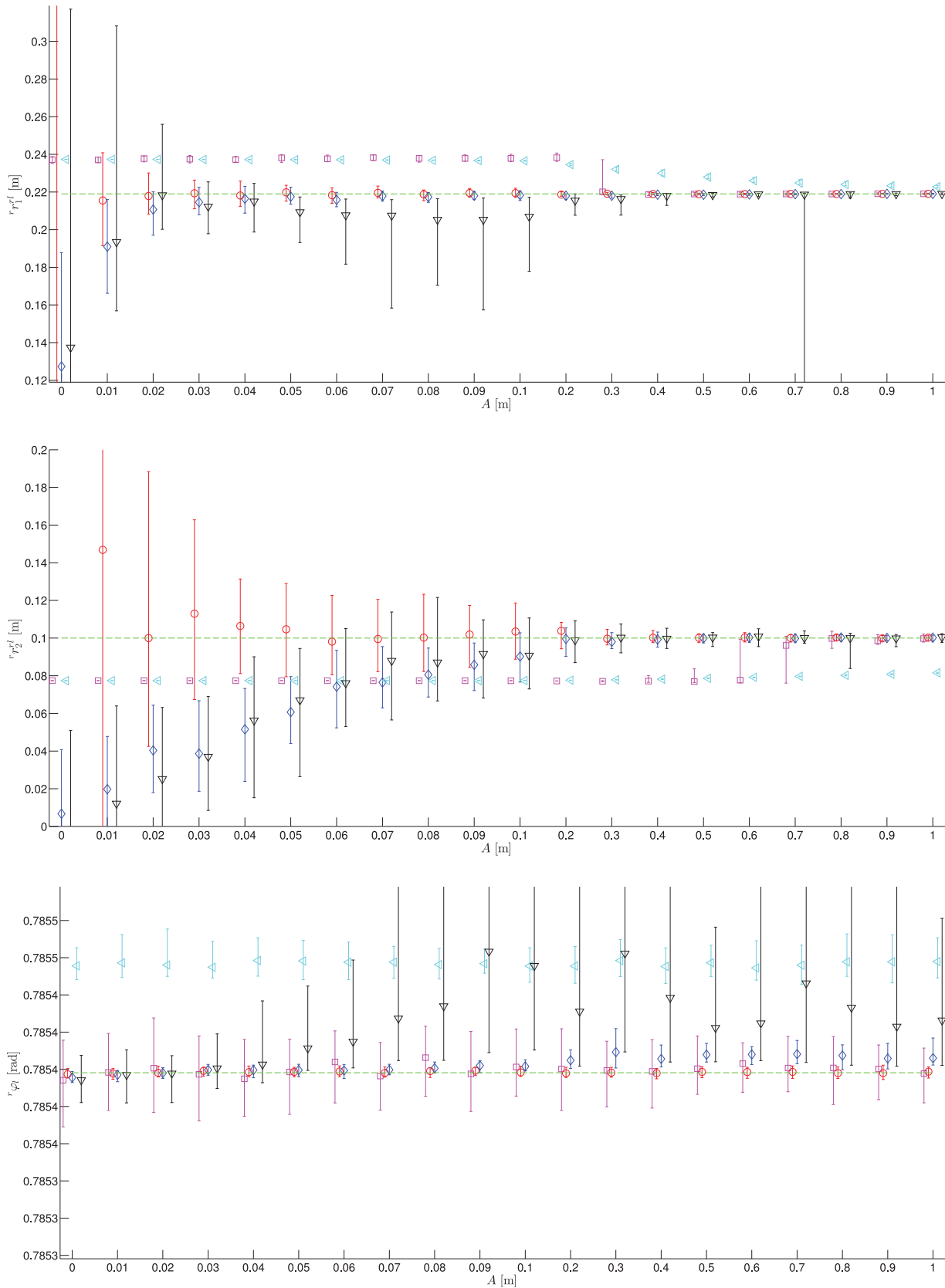


Fig. 7. Evolution of the calibration parameters at different sinusoid amplitudes for LM (red circles), EKF with tuned prior (blue diamonds), EKF with tight prior (cyan triangles), EKF with loose prior (black triangles), and our online batch algorithm (magenta squares). The green lines represent the ground truth parameters, the markers the medians, and the error bars the first and third quantiles.

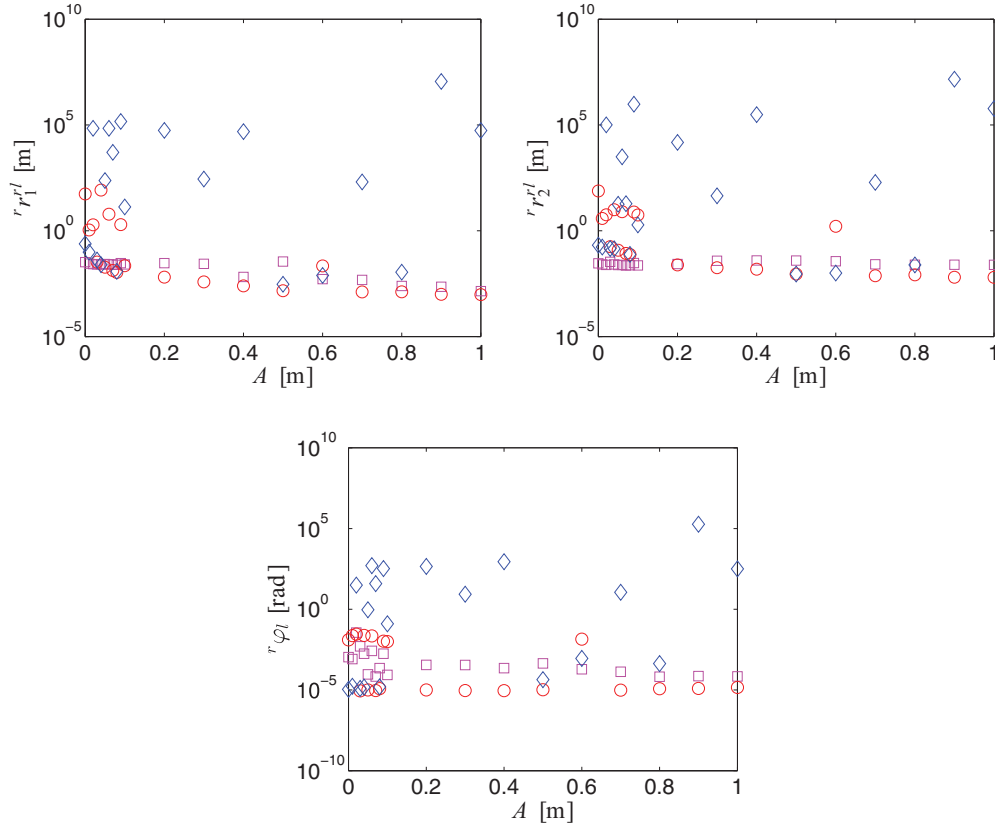


Fig. 8. Maximum absolute errors for the calibration parameters at different sinusoid amplitudes for LM (red squares), EKF with tuned prior (blue diamonds), and our online batch algorithm (magenta circles).

While our online batch method leaves the unobservable parameters at their initial guess, the LM and EKF perform poorly in that situation. In fully observable conditions (from $A \approx 0.5$ [m]), the three methods yield similar results, but our online batch algorithm uses on average less than 50% of the available data. It can be noted that we have chosen a conservative numerical rank threshold so that the parameters get updated only if there is enough confidence in the data. As we discount data in comparison to other methods, the estimates produced by our algorithm have a slightly larger spread.

One key question that arises when looking at Figure 7 is: *what behavior is preferable for a calibration algorithm?* In cases that are completely unobservable or weakly observable, the EKF and batch algorithms make a terrible mistake; both algorithms fit to the noise, resulting in widely varying estimates for the unobservable calibration parameters. This happens for different reasons. The EKF simply takes a random walk, fitting to noise in each step. The magnitude of this walk is determined by the strength of the prior. The batch algorithm computes the low curvature of the cost space along these axes and takes large jumps towards physically implausible values that minimize the cost function.

For the estimates produced by the batch algorithm, the median is mostly near to the true value, but the variability is absolutely undesirable for an online, safety-critical

system (see for instance Figure 8 that displays the maximum absolute errors for the calibration parameters). The three variations of the EKF show the difficulty in tuning the parameters for an application so that it will converge without biasing the results. In many papers about self-calibration, authors suggest to add a simple zero-velocity motion model to calibration parameters to allow them to change in the EKF over time. We have not done that for these experiments, but we believe that this represents another extremely difficult tuning process: getting the motion model just right so that it allows slow variation in the parameters without allowing excessive drift.

In contrast, our algorithm, behaves exactly as we would like; it listens to the incoming data, only updating parameters when it is convinced that the evidence is *signal* and not noise. In all cases, our algorithm produces low variance results, either around the initial guess, or near the true value.

4.1.4. Real-world experiments. In order to validate our method on real-world data, we have used the “Lost in the Woods Dataset” provided with the courtesy of Tim Barefoot (Tong et al., 2012). This dataset contains approximately 20 minutes of a robot driving amongst a forest of tubes which serve as landmarks. The ground truth comes from a motion capture system that tracks robot motion and tube locations. For the calibration parameters, we have only

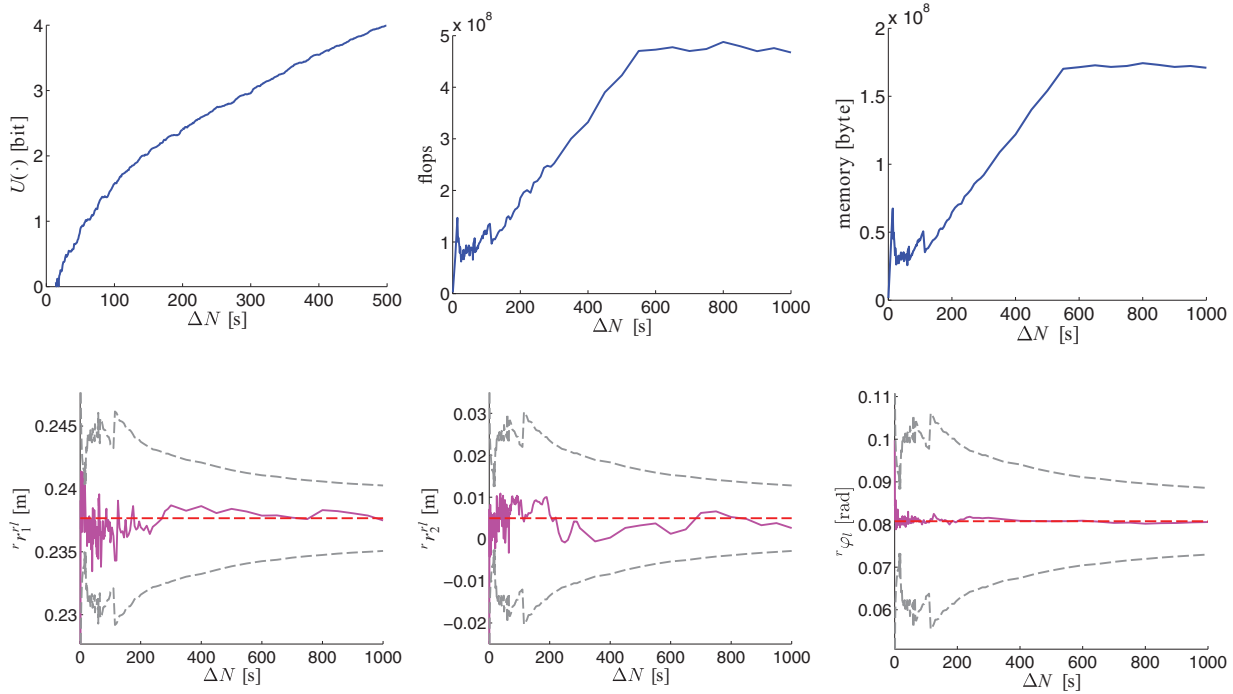


Fig. 9. Online parameters analysis. The top-left plot shows the information gain, the top-middle plot the maximum number of flops, the top-right plot the maximum memory footprint, and the bottom plots the online batch calibration estimates (magenta), with their 3- σ bounds (gray) and LM (red), as a function of batch sizes.

access to $r_1^{rl} = 0.219$ [m] that was roughly measured with a tape. We assume the others are implicitly set to 0 (i.e. $r_2^{rl} = 0$ [m] and $r_{\phi_l} = 0$ [rad]).

In a first effort, we investigate the influence of the online parameter set, \mathcal{P}_O , on the behavior of our algorithm. The aim of this experiment is to determine adequate values for these parameters. As we pointed out in Section 3.5 and as can be seen in the top-left plot of Figure 9, the information gain threshold, $\delta_{\mathcal{U}}$, and the batch size, ΔN , are coupled. Here, we have first inserted a large batch of 150 [s] into our estimator and subsequently computed the information gain as a function of newly inserted batches of varying sizes. This plot shows that larger batch sizes yield higher information gain and provides a way of selecting $\delta_{\mathcal{U}}$ for a given ΔN . Using this relation, we have run our algorithm with varying batch sizes and we show in the top-middle and top-right plot of Figure 9 the maximum number of flops and memory footprint, and in the bottom plots the final calibration estimates, along with their 3- σ bounds plotted around the full-batch LM output. In the absence of accurate ground truth, we can consider this latter as the gold standard estimate. These plots provide a guideline for choosing the online parameters, i.e. we require low memory footprint, a minimal number of flops, accuracy with respect to LM, and precision.

From the parameter analysis, we select a batch size of $\Delta N = 30$ [s] and an information gain threshold of $\delta_{\mathcal{U}} = 0.2$ [bit], and we display in Figure 10 the qualitative results of our online batch algorithm. Using less than 50% of the measurements, we could recover accurate landmark

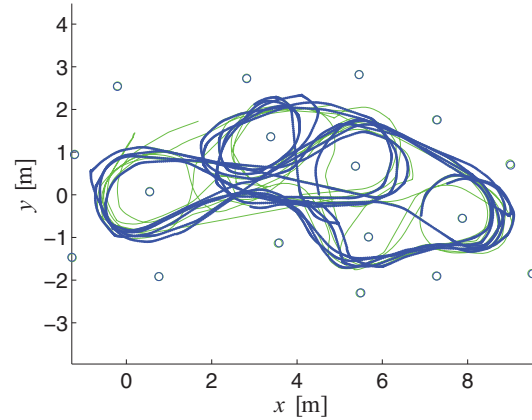


Fig. 10. Qualitative results for the estimation of robot's poses and landmark positions for the "Lost in the Woods Dataset". The ground truth trajectory, ${}^wT_{r_{1:N}}$, is shown in green, ground truth landmark positions, ${}^w\ell_{1:N}$, in green circles, trajectory estimate, ${}^w\hat{T}_{r_{1:N}}$, in blue, and landmark positions estimate, ${}^w\hat{\ell}_{1:N}$, in blue circles.

positions, robot's poses for the informative batches, and calibration parameters. We show in Figure 11 the evolution of the calibration parameters as a function of the incoming batches with a comparison to LM and EKF. From this plot, we observe that the initial guess for wT_l from the dataset was relatively correct. The dataset being well-behaved, i.e. all of the parameters are observable, the three methods give comparable results, as was already noticed in the simulation experiments. The higher variability in the second

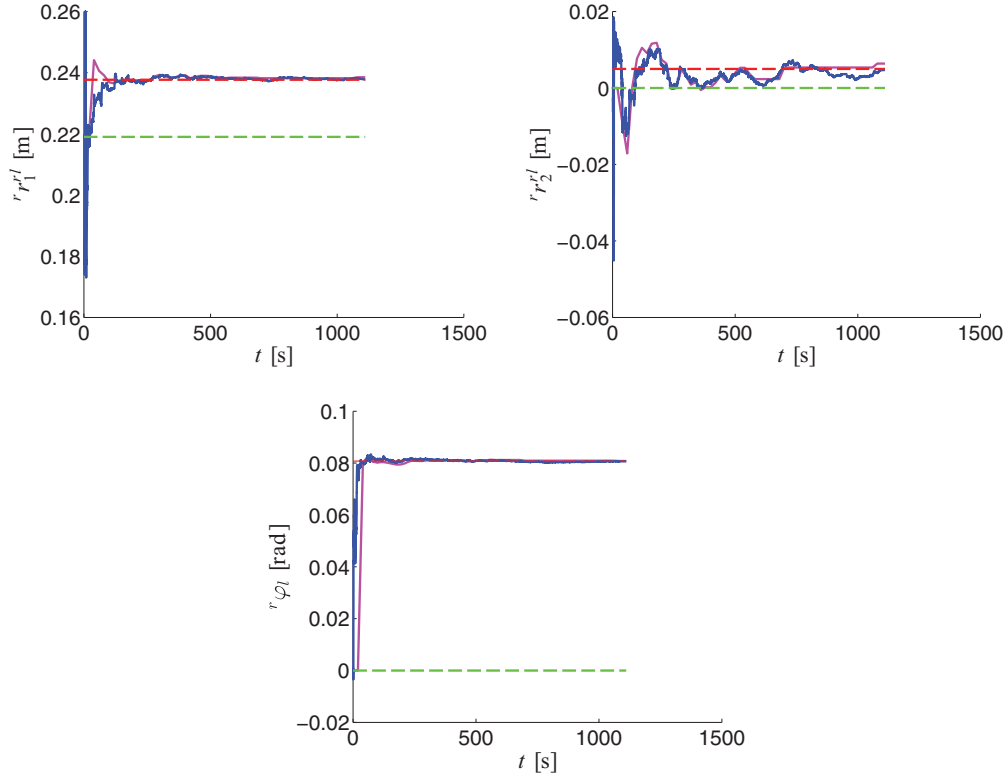


Fig. 11. Evolution of the calibration parameter estimates, $\hat{\mathbf{T}}_l = (r_1^{rl}, r_2^{rl}, \phi_l)$, as a function of the incoming batches for the “Lost in the Woods Dataset”. The initial guess, $\mathbf{T}_l^{(0)}$, is shown in green, the estimates for LM in red, for EKF in blue, and for our online batch algorithm in magenta.

translational component, r_2^{rl} , was also observed in Figure 7. We suspect that the lack of lateral motion of the robot renders it weakly observable.

4.2. Incremental calibration of a visual-inertial sensor unit

In this experiment we considered the extrinsic calibration of a visual-inertial sensor unit, which consisted of two MT9V034 global shutter cameras in a plan-parallel configuration and an ADIS16448 IMU (see Figure 12 and an extended description of the device in Nikolic et al. (2014)). We fixed the unit on a swivel mount, which allowed rotations around the vertical axis. The mount was attached to a sliding element on a track that allowed horizontal translations along a single axis. An AprilTag calibration pattern board was placed in front of the setup in order to provide a set of fixed landmarks for the cameras to observe while in motion (Furgale et al., 2013a; Olson, 2011). With the sensor unit fixed to the mount, we collected data while performing two different motion patterns. Then, we released the unit from the mount and performed the third motion pattern. These motion patterns occur in three consecutive phases, as follows:

- (1) during the first phase, the unit was motionless;
- (2) during the second phase, the mounted unit was rotated around the vertical axis and translated along the horizontal axis; and

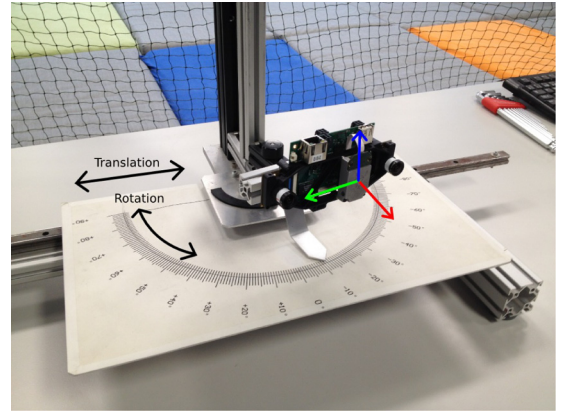


Fig. 12. Experimental setup for the incremental calibration of the visual-inertial sensor unit. The unit, consisting of two global shutter cameras and an IMU, was placed on a swivel mount, which was attached to a vertical shaft. This allowed horizontal translations and rotations around the vertical axis. The axis diagram indicates the x , y and z directions (red, green and blue, respectively) in the IMU’s frame of reference.

- (3) during the third phase, the unit was released from the mount and moved freely, performing unconstrained six-degree-of-freedom motion.

Each phase lasted approximately 60 [s]. The motions during the second and third phases were created by moving the

device by hand. We repeated the experiment a total of three times, resulting in three different datasets. The initial calibration parameters were obtained via hand measurements.

The goal of this experiment was to incrementally estimate the relative transformations between both cameras and the IMU (12 parameters). We used the continuous-time motion estimation and calibration algorithm described in Furgale et al. (2013a) and available as a calibration toolbox at <https://github.com/ethz-asl/kalibr>. This toolbox was also used to calibrate the intrinsics of each camera before the experiment. As auxiliary variables, the algorithm estimates the time-varying pose of the IMU, time varying IMU biases, the pose of the pattern board, and the two-degree-of-freedom gravity direction (assuming known gravity magnitude). The successive motion patterns, no motion followed by planar motion, represent cases in which it is known analytically that the calibration parameters are not fully observable (Kelly and Sukhatme, 2011; Mirzaei and Roumeliotis, 2008). In the final phase, full six-degree-of-freedom motion, all calibration parameters become observable. Our incremental calibration framework knows nothing about these three phases, and must automatically infer the time-varying observability of each parameter from the data. To the best of the authors' knowledge, this is the first paper that attempts such task.

In order to select the parameters of the algorithm, we divided the data into a training set and a test set. We used the first dataset for training, and the second and third datasets for testing. During training, we tuned the parameters of the algorithm, i.e. the batch size, ΔN , the information gain threshold, $\delta_{\mathcal{U}}$, and the numerical rank thresholds, ϵ_{θ} and ϵ_{ψ} . The tuning procedure was as follows. First, using a $\Delta N = 10$ [s] batch taken from the motionless phase, we started with a small ϵ_{ψ} and increased it until the Gauss–Newton algorithm converged without many failed steps. This gave $\epsilon_{\psi} = 1$. Second, we reduced the batch size, ΔN , while monitoring the singular values on a different batch (also taken from the motionless phase), until the gap between the observable and non-observable directions started to become less apparent. This produced $\Delta N = 5$ [s]. Third, we compared the singular values in two 5 [s] batches from the motionless and planar motion phases, and selected ϵ_{θ} as low as possible while still differentiating between strongly and weakly observable directions in parameter space. This resulted in $\epsilon_{\theta} = 5000$. Last of all, we selected the information gain threshold $\delta_{\mathcal{U}}$ as roughly 1% of the peak information gain during the motionless phase. This yielded $\delta_{\mathcal{U}} = 1$ [bit].

Having selected the parameters, we ran the incremental calibration algorithm on the testing data. Figure 13 shows the results of the calibration run on both the testing and training datasets. The top panel shows the information gain, in bits, produced by each 5 [s] batch. The middle panel shows the total number of batches that pass the information gain test since the start of the run. The bottom panel shows the rank deficiency of the parameters, $K - r_{\epsilon_{\theta}}$ (solid lines), and of the auxiliary variables, $L - r_{\epsilon_{\psi}}$ (dashed

lines). Different colors correspond to different datasets. The dotted vertical lines indicate the phase changes.

The results depicted in Figure 13 clearly demonstrate the potential of this algorithm for online calibration where the observability of all variables is not guaranteed at all times. As well as locking down unobservable directions in parameter space, the algorithm provides nullspace basis vectors that a developer can use to understand which parameters are unobservable to gain intuition about the specific calibration problem.

Note the behavior of the rank deficiency curves in Figure 13. During the first phase, the camera-to-camera transformation is estimated from keypoint correspondences (six-degree-of-freedom) and known geometry of the checkerboard. However, since the unit is motionless, there is no information available to correlate image keypoints to IMU measurements. Hence the parameter rank deficiency is six, which corresponds to the three translation and three rotation parameters of the unobservable camera-system-to-IMU transformation. The planar motion phase starts 60 [s] later. In this phase most of these parameters become observable, except one. This is reflected by the rank deficiency of the parameters, which drops to 1 at $t = 60$ [s]. This basis vector for this rank deficiency aligns with translation along the normal of the plane that the sensor head is constrained to. The rank deficiency finally becomes zero at around $t = 120$ [s], when the 6-DoF motion excites all axes of the camera-to-IMU transformations, rendering the remaining parameter observable. The final drop occurs at slightly different times for different datasets, namely at $t = 120$, 125 and 130 [s]. This is because the six-degree-of-freedom motion phase starts 5 and 10 [s] late in the first and second datasets, respectively: the time-delayed spikes in the information gain (top panel) reflect this.

The rank deficiency of the auxiliary variables follows a similar trend. At the beginning of the experiment, there are a total of seven or eight unobservable directions: the global six-degree-of-freedom pose of the sensor unit, and a two-dimensional subspace consisting of linear combinations of accelerometer biases and the direction of the gravity vector. The magnitude of the gravity vector is not estimated in this experiment, only its direction. Hence the rank deficiency of two and not three. In the second phase, the planar motion of the unit renders the gravity direction observable. The global pose, however remains unobservable throughout the rest of experiment. This is consistent with the fact that there is no absolute position information.

5. Conclusion

In this paper, we have investigated the development of an automated method that deals with degenerate configurations occurring during calibration runs. By means of matrix analysis, our algorithm identifies unobservable directions in the calibration parameter space and locks them at their initial guess until they become fully observable. To leverage

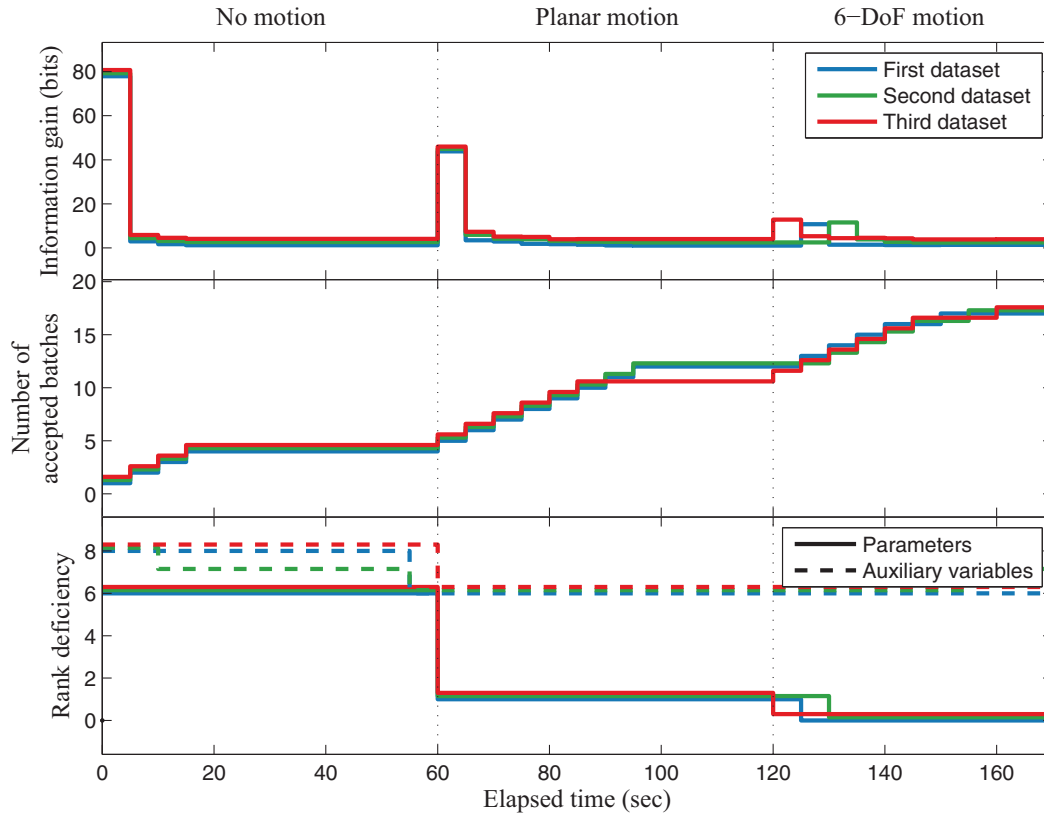


Fig. 13. Results of the incremental calibration algorithm on the visual-inertial calibration problem. The top panel shows the information gain, in bits, produced by each batch. The middle panel shows the total number of accepted batches since the start of the calibration. The bottom panel shows the rank deficiency of the parameters (solid lines), and of the auxiliary variables (dashed lines). Different colors correspond to different datasets. The dotted vertical lines indicate the phase changes.

the otherwise intractable computational and space complexity of nonlinear optimization methods, we have proposed an information-theoretic scheme that evaluates the utility of data sample batches for the calibration.

5.1. Contributions

The first contribution of this work concerns an online observability measure of the parameter space. Whilst the majority of state-of-the-art approaches are either not aware of singularities or perform algebraic observability analysis, we have presented a method based on rank-revealing QR and SVDs of the Fisher information matrix that separates the calibration parameters into observable and unobservable subspaces. Splitting up the parameter space, our algorithm only updates the observable dimensions, leaving the rest intact. Along these lines, we have developed the concept of numerical observability, the counterpart of numerical rank in linear algebra, as opposed to the purely algebraic view in control theory. Through our simulated experiments, we have illustrated the difficulty in assessing observability in presence of noisy sensor data and shown the robustness of our algorithm to properly execute this task in comparison to other methods.

In a second effort, we have validated the theoretical intuitions on the behavior of the entropy difference between prior and posterior distributions when new data batches enter the estimator through simulated and real-world experiments. Linked to the observability, this measure perfectly fulfills our expectations for the selection of informative data and is hence a sound complement to nonlinear least-squares methods.

5.2. Discussion and future work

The presented algorithm for the online self-calibration of robotic sensors has several limitations and therefore offers vast opportunities for further research directions that we shall discuss in the rest of this section.

The key limitation in the current implementation lies in the assumption that the calibration parameters remain constant during the calibration run. For the proper deployment of our method into consumer-level sensors, we must be able to cope with slowly or abruptly varying parameters. As an example, we can consider an autonomous vehicle with cameras embedded into the side mirrors. Whenever the user opens or closes the doors, the relative poses of the cameras might change due to the mechanical shock. It is

nevertheless possible to detect these cases by monitoring sudden variations in the prediction error of the estimator and restarting the calibration algorithm from scratch. A probably smarter approach would be to add a forgetting factor to our algorithm. Using this strategy, the forgetting factor would give exponentially less weight to older data batches, discarding them eventually, to favor fresh information. The covariance matrix of the estimates should be scaled accordingly to avoid a too strong convergence of our online algorithm, leading to discounting any novel data.

A current limitation of our algorithm is the dependence on a reasonable initial guess for the calibration parameters. To reach the fully unsupervised goal we have in mind, the calibration algorithm should be able to bootstrap itself from scratch. Along the same lines, our implementation uses Gauss–Newton update steps in the nonlinear optimization, which is hence only guaranteed to converge if the initial guess is close enough to a local minimum. To withdraw this assumption, we suggest a trust-region method such as Powell’s dog-leg algorithm (Powell, 1970). This class of methods uses a combination of gradient descent steps for global convergence and Gauss–Newton steps for local convergence.

Although the results from our preliminary experiments are promising, there are a number of open questions regarding the tuning of the numerical rank thresholds, ϵ_θ and ϵ_ψ . At the moment, we find the right combination of ϵ_θ and ϵ_ψ by working on a training dataset, and inspecting the singular values and the resulting rank deficiencies. However, the tuning process is not always intuitive due to parameter inter-dependencies. Namely, the magnitude of the singular values changes with the level of noise and with the size of ΔN the batches, and ϵ_ψ indirectly affects the singular values of A_θ via the RRQR, since A_θ is a function of Q_1 (see Section 3.3). This is undesirable. As future work, we plan on:

- (a) defining a principled, intuitive approach to selecting the tuning parameters; and
- (b) developing a method for automated tuning.

Be that as it may, we have witnessed that these thresholds only need to be calculated once per sensor setup, which remains a manageable task for an expert before the deployment of the algorithm in the field. Skimming through the literature, we have observed that even experts (Davis, 2011) in numerical analysis employ a heuristic that “has worked well in practice”. It is important to recall here that an incorrect numerical rank detection can lead to utterly wrong estimates as was demonstrated in Section 4. In that regard, we suggest a conservative approach, such that these thresholds may underestimate the numerical rank rather than overestimating it.

In the presence of outliers or other effects that violate the assumption of normality, we believe we can adopt an M-estimator framework to smoothly dampen the effect of outliers in the estimation process. This is motivated by the

simplicity at which it can be integrated into a nonlinear optimization method. However, whenever the initial guess is not close enough to a local minimum, the M-estimator may incorrectly consider data points as being outliers and thus dramatically bias the optimization. We may address this issue by adapting the M-estimator at each iteration, i.e. being conservative at the beginning and stricter as the algorithm converges to a local minimum. If the sensor data is absolutely not normally distributed, our method may fail and would require an ad hoc modification, e.g. accounting for multimodal distributions.

Finally, we will begin the work of integrating this algorithm into our complicated robotics systems for navigating in dynamic urban environments (Furgale et al., 2013b; Kummerle et al., 2013) where safety and robustness are of the highest priority.

Funding

This work is supported in part by the European Community’s Seventh Framework Programme (FP7/2007-2013; grant numbers #269916 (V-Charge) and #610603 (EUROPA2)).

Notes

1. Informally, observability means that the parameters can be inferred unambiguously from some local batch of measurement data (Jazwinski, 1970).
2. See Lébraly et al. (2010) for a handy table of degenerate cases or Kim and Chung (2006) for a more theoretical analysis.
3. Informally, the rank of a matrix corresponds to the maximum number of linearly independent column or row vectors.

References

- Bishop CM (2006) *Pattern Recognition and Machine Learning*. New York: Springer.
- Brookshire J and Teller S (2011) Automatic calibration of multiple coplanar sensors. In: *Proceedings of the robotics: science and systems conference (RSS)*.
- Brookshire J and Teller S (2012) Extrinsic calibration from per-sensor egomotion. In: *Proceedings of the robotics: science and systems conference (RSS)*.
- Censi A, Marchionni L and Oriolo G (2008) Simultaneous maximum-likelihood calibration of odometry and sensor parameters. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.
- Chan TF (1987) Rank revealing QR factorizations. *Linear Algebra and its Applications* 88–89: 67–82.
- Chan TF and Hansen PC (1992) Some applications of the rank revealing QR factorization. *SIAM Journal on Scientific and Statistical Computing* 13(3): 727–741.
- Davis TA (2011) Algorithm 915: SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization. *ACM Transactions on Mathematical Software* 38(1): 1–24.
- Fiore PD (2001) Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(2): 140–148.
- Furgale P, Rehder J and Siegwart R (2013a) Unified temporal and spatial calibration for multi-sensor systems. In: *Proceedings of the*

- the IEEE/RSJ international conference on intelligent robots and systems (IROS).
- Furgale P, Schwesinger U, Rufli M, Derendarz W, Grimm H, Mühlheller P, et al (2013b) Toward automated driving in cities using close-to-market sensors, an overview of the V-Charge project. In: *Proceedings of the IEEE intelligent vehicles symposium (IV)*.
- Gao C and Spletzer JR (2010) On-line calibration of multiple LIDARs on a mobile vehicle platform. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.
- Gelman A, Carlin JB, Stern HS and Rubin DB (2003) *Bayesian Data Analysis*. London: Chapman and Hall/CRC.
- George A and Ng E (1988) On the complexity of sparse QR and LU factorization of finite-element matrices. *SIAM Journal on Scientific and Statistical Computing* 9(5): 849–861.
- Golub GH and Loan CFV (1996) *Matrix Computations*. Baltimore, MD: The Johns Hopkins University Press.
- Hansen PC (1987) The truncated SVD as a method for regularization. *BIT Numerical Mathematics* 27(4): 534–553.
- Hansen PC (1988) The 2-norm of random matrices. *Journal of Computational and Applied Mathematics* 23(1): 117–120.
- Hansen PC (1998) *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).
- Harris C and Stephens M (1988) A combined corner and edge detector. In: *Proceedings of the 4th Alvey vision conference*.
- Harrison A and Newman P (2011) TICSyn: Knowing when things happened. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, pp. 356–363.
- Heng L, Li B and Pollefeys M (2013) CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*.
- Jauffret C (2007) Observability and Fisher information matrix in nonlinear regression. *IEEE Transactions on Aerospace and Electronic Systems* 43(2): 756–759.
- Jazwinski AH (1970) *Stochastic Processes and Filtering Theory*. New York: Academic Press.
- Kelley C (1999) *Iterative Methods for Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).
- Kelly J, Roy N and Sukhatme G (2014) Determining the time delay between inertial and visual sensor measurements. *IEEE Transactions on Robotics* 30(6): 1514–1523.
- Kelly J and Sukhatme GS (2011) Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research* 30(1): 56–79.
- Kim JH and Chung MJ (2006) Absolute motion and structure from stereo image sequences without stereo correspondence and analysis of degenerate cases. *Pattern Recognition* 39: 1649–1661.
- Kummerle R, Grisetti G and Burgard W (2011) Simultaneous calibration, localization, and mapping. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*.
- Kummerle R, Ruhnke M, Steder B, Stachniss C and Burgard W (2013) A navigation system for robots operating in crowded urban environments. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*. DOI:10.1109/ICRA.2013.6631026.
- Lébraly P, Royer E, Ait-Aider O and Dhome M (2010) Calibration of non-overlapping cameras - application to vision-based robotics. In: *Proceedings of the British machine vision conference (BMVC)*.
- Levinson J and Thrun S (2010) Unsupervised calibration for multi-beam lasers. In: *Proceedings of the international symposium on experimental robotics (ISER)*.
- Li M and Mourikis AI (2014) Online temporal calibration for camera-IMU systems: Theory and algorithms. *The International Journal of Robotics Research* 33(7): 947–964.
- Lindley DV (1956) On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics* 27(4): 879–1212.
- Maddern W, Harrison A and Newman P (2012) Lost in translation (and rotation): Rapid extrinsic calibration for 2D and 3D LIDARs. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.
- Mair E, Fleps M, Suppa M and Burschka D (2011) Spatio-temporal initialization for IMU to camera registration. In: *2011 IEEE international conference on robotics and biomimetics (ROBIO)*. IEEE, pp. 557–564.
- Martinelli A, Scaramuzza D and Siegwart R (2006) Automatic self-calibration of a vision system during robot motion. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.
- Maybank SJ and Faugeras OD (1992) A theory of self-calibration of a moving camera. *International Journal of Computer Vision* 8(2): 123–151.
- Maye J, Furgale P and Siegwart R (2013) Self-supervised calibration for robotic systems. In: *Proceedings of the IEEE intelligent vehicles symposium (IV)*.
- Menq CH, Born JH and Lai JZ (1989) Identification and observability measure of a basis set of error parameters in robot calibration. *Journal of Mechanisms, Transmissions, and Automation in Design* 111: 513–518.
- Mirzaei FM and Roumeliotis SI (2008) A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics* 24(5): 1143–1156.
- Nikolic J, Rehder J, Burri M, Gohl P, Leutenegger S, Furgale PT and Siegwart R (2014) A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM. In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE.
- Olson E (2011) AprilTag: A robust and flexible visual fiducial system. In: *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, pp. 3400–3407.
- Powell MJ (1970) A new algorithm for unconstrained optimization. In: Rosen J, Mangassarian O and Ritter K (eds.) *Non-linear Programming*. New York: Academic Press, pp. 31–65.
- Rehder J, Beardsley P, Siegwart R and Furgale P (2014) Spatio-temporal laser to visual/inertial calibration with applications to hand-held, large scale scanning. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Chicago, IL, USA, pp. 459–465.
- Rufli M, Scaramuzza D and Siegwart R (2008) Automatic detection of checkerboards on blurred and distorted images. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*.
- Sheehan M, Harrison A and Newman P (2010) Automatic self-calibration of a full field-of-view 3D n-laser scanner. In: *Proceedings of the international symposium on experimental robotics (ISER)*.

- Sheehan M, Harrison A and Newman P (2012) Self-calibration for a 3D laser. *The International Journal of Robotics Research* 31(5): 675–687.
- Sturm PF and Maybank SJ (1999) On plane-based camera calibration: A general algorithm, singularities, applications. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.
- Tarantola A (2005) *Inverse Problem Theory and Methods for Model Parameter Estimation*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).
- Teichman A, Miller S and Thrun S (2013) Unsupervised intrinsic calibration of depth sensors via SLAM. In: *Proceedings of the robotics: science and systems conference (RSS)*.
- Tong CH, Furgale P and Barfoot TD (2012) Gaussian process Gauss–Newton: Non-parametric state estimation. In: *Proceedings of the conference on computer and robot vision (CRV)*.
- Underwood J, Hill A and Scheduling S (2007) Calibration of range sensor pose on mobile platforms. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*.
- Zhang L, Liu Z and Xia C (2002) Clock synchronization algorithms for network measurements. In: *Proceedings of the 21st annual joint conference of the IEEE Computer and Communications Societies*, vol. 1, pp. 160–169.
- Zhang Q and Pless R (2004) Extrinsic calibration of a camera and laser range finder (improves camera calibration). In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)*.
- Zhang Z (1999) Flexible camera calibration by viewing a plane from unknown orientations. In: *Proceedings of IEEE international conference on computer vision (ICCV)*.