# A fast and accurate approximation for planar pose graph optimization

**Article** *in* The International Journal of Robotics Research · May 2014

DOI: 10.1177/0278364914523689



**4 authors:**

Luca Carlone
Massachusetts Institute of Technology
**146** PUBLICATIONS **5,146** CITATIONS

SEE PROFILE

Rosario Aragues
University of Zaragoza
**52** PUBLICATIONS **614** CITATIONS

SEE PROFILE

Jose A Castellanos
University of Zaragoza
**68** PUBLICATIONS **3,211** CITATIONS

SEE PROFILE

Basilio Bona
Politecnico di Torino
**138** PUBLICATIONS **1,942** CITATIONS

SEE PROFILE

# A Fast and Accurate Approximation for Planar Pose Graph Optimization

Luca Carlone, Rosario Aragues, José A. Castellanos, Basilio Bona

***Abstract*—This work investigates the *pose graph optimization* problem, which arises in maximum likelihood approaches to *Simultaneous Localization And Mapping* (SLAM). State-of-the-art approaches have been demonstrated to be very efficient in medium and large-sized scenarios; however, their convergence to the maximum likelihood estimate heavily relies on the quality of the initial guess. We show that, in planar scenarios, pose graph optimization has a very peculiar structure. The problem of estimating robot orientations from relative orientation measurements is a quadratic optimization problem (after computing suitable *regularization terms*); moreover, given robot orientations, the overall optimization problem becomes quadratic. We exploit these observations to design an approximation of the maximum likelihood estimate, which does not require the availability of an initial guess. The approximation, named LAGO (*Linear Approximation for pose Graph Optimization*), can be used as a stand-alone tool or can bootstrap state-of-the-art techniques, reducing the risk of being trapped in local minima. We provide analytical results on existence and sub-optimality of LAGO, and we discuss the factors influencing its quality. Experimental results demonstrate that LAGO is accurate in common SLAM problems. Moreover, it is remarkably faster than state-of-the-art techniques, and is able to solve very large-scale problems in few seconds.**

***Index Terms*—Pose Graph Optimization, Simultaneous Localization And Mapping, Mobile Robots, Graph Theory, Linear Estimation.**

## I. Introduction

WHEN a robot travels in an unknown scenario, the construction of a world model and the concurrent estimation of its location is crucial for mission accomplishment, to enhance motion planning effectiveness, and to attain a desired level of situational awareness. For this reason, Simultaneous Localization and Mapping (SLAM) has been a central topic in robotic research (see the survey papers [19], [20]). SLAM approaches can be distinguished into two categories: *online* (or *incremental*) and *batch* techniques. The former includes techniques that incrementally estimate the SLAM posterior by recursively including the most recent measurements in the estimate. This is typically the case of a robot that acquires measurements from the environment and, at each time step, includes such information in the estimate. Batch SLAM approaches, instead, compute the estimate taking into account at the same time all the available measurements; this solution may be required, for instance, when the data, acquired during robot operation, have to be processed off-line to produce a meaningful representation of the scenario.

*Pose graph optimization* has been demonstrated to be an effective formulation of batch SLAM, allowing, for instance, to easily introduce absolute position information (i.e., GPS) or to recover from bad loop closings. The objective of pose graph optimization is to estimate the poses assumed by a mobile robot at subsequent instants of time (*pose graph*) from the available measurements. The computation of the maximum likelihood estimate of robot poses leads to a hard non-convex optimization problem, with multiple local minima.

*Related work*: In the seminal paper [38], Lu and Milios propose the use of nonlinear optimization to estimate robot poses. Thrun and Montemerlo [53] extend this framework to the general case of a graph containing both robot poses and landmark positions. Konolige [33] restricts the optimization to the poses involved in at least one loop closing, providing a remarkable computational advantage for graphs with low connectivity. Frese *et al.* [21] propose a multilevel relaxation approach. A further breakthrough in the literature of graph-based approaches is the use of *incremental pose parametrization*, proposed in [43]. Olson *et al.* show how a clever selection of the optimization variables can greatly simplify problem structure, enabling fast computation. Grisetti *et al.* [23], [24] extend this framework, taking advantage of stochastic gradient descent in planar and three-dimensional scenarios. Kaess *et al.* [29]–[31] present an elegant formalization of SLAM using a *Bayes tree* model and investigate incremental estimation techniques. Kuemmerle *et al.* [35] introduce the g2o framework for solving general optimization problems with variables belonging to manifolds. Olson and Agarwal [42], and Sünderhauf and Protzel [51], [52] propose techniques to enhance robustness in presence of outliers. These methods include also the topology of the graph in the optimization process, activating or deactivating constraints with suitable *switch* variables [51], [52], or using a *max*-mixture model [42]. Rosen *et al.* [46] propose a trust-region method to enhance convergence of iterative optimization.

The mentioned techniques are iterative, in the sense that, at each iteration, they solve a local convex approximation of the original problem, and use such local solution to update the estimate. This process is then repeated until the optimization variable converges to a minimum of the cost function. Some

This work was partially funded by Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) under MEMONET National Research Project, by projects Ministerio de Ciencia e Innovacion DPI2009-08126, DPI2009-13710, DPI2012-36070, and grant MEC BES-2007-14772.

L. Carlone is with the College of Computing, Georgia Institute of Technology, 30332 Atlanta, USA, luca.carlone@gatech.edu

B. Bona is with the Dipartimento di Automatica e Informatica, Politecnico di Torino, 10129 Torino, Italy, basilio.bona@polito.it

R. Aragues is with Clermont Université, Institut Pascal, BP 10448, F-63000 Clermont-Ferrand, France, and with CNRS, UMR 6602, IP, F-63171 Aubiere, France, mdarague@univ-bpclermont.fr

J.A. Castellanos is with the Departamento de Informática e Ingeniería de Sistemas, Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, E-50018 Zaragoza, Spain, jacaste@unizar.es

original attempts to exploit the mathematical structure of pose graph optimization have been recently proposed in [17], [18], [28], [45], [55]. Rizzini [45] presents an interesting closed-form solution, under the assumption that measurement covariances are identity matrices. However, the closed-form estimate requires to solve a system of multivariate quadratic polynomial equations and it is not clear if the approach admits solution in general (Section III-B in [45]), and what is its computational cost. Dubbelman *et al.* [17], [18] propose a closed-form estimate for graphs with a single loop and isotropic noise. Huang *et al.* [28] discuss the convexity properties of SLAM, drawing conclusions about the importance of the orientation measurements, that are also confirmed by the present paper. More recently, Huang *et al.* [55] investigate the number of minima in optimization-based SLAM.

*Contribution and paper outline*: Although state-of-the-art approaches have been demonstrated to produce impressive results on real world problems, their iterative nature requires the availability of an initial guess for nonlinear optimization: convergence to a global minimum of the cost function cannot be guaranteed in general, and, if the initial guess is outside the region of attraction of the global optimum, the iterative process is likely to be stuck in a local minimum (*orientation wraparound* problem). The main contribution of this work is a fast an accurate approximation for pose graph optimization that requires no initial guess. The approximation can be used as a stand-alone tool or can bootstrap iterative techniques.

The paper is organized as follows. Section II reviews the usual formulation of pose graph optimization and recalls the basic principles underlying iterative optimization techniques.

Section III shows that, in planar scenarios, pose graph optimization has a very peculiar structure. Assuming that the relative position and relative orientation measurements are independent, the problem of estimating robot orientations from relative orientation measurements is a quadratic problem; moreover, given robot orientations, the overall optimization problem becomes quadratic.

In Section IV we exploit these observations to obtain an approximation of the maximum likelihood solution. The corresponding algorithm is named LAGO (*Linear Approximation for pose Graph Optimization*), where the term "linear" refers to the fact that the original nonlinear problem is approximated by a sequence of two linear estimation problems. The approximation is computed in three phases: we first obtain an estimate for robot orientations, which are the source of nonlinearity in the optimization problem. In the second phase we use such estimate to express the relative position measurements in a global frame. In the third phase, we solve a linear estimation problem involving robot poses.

Section IV also provides an assessment of LAGO. We prove the existence and uniqueness of the proposed approximation. Moreover, we show that the linear approximation is equivalent to a Gauss-Newton step around a suboptimal estimate; we provide an analytical upper bound of the suboptimality gap (difference between the objective attained by the suboptimal solution and the optimal objective value) of such suboptimal estimate. As a conclusion of the section we present a probabilistic interpretation of the proposed approach.

Section V goes back to orientation estimation. We show that, in order to frame orientation estimation as a quadratic problem, it is necessary to include integer variables (regularization terms). We use graph theoretical concepts to design an algorithm that finds these integer variables.

Section VI describes an implementation of the proposed approach and reports several results from simulations and real tests. We compare a c++ implementation of LAGO (available at [7]) with state-of-the-art techniques, namely, Toro [24], g2o [35], and iSAM [30]. Experimental evidence demonstrates that LAGO is very accurate in common SLAM problems and it is remarkably faster than state-of-the-art techniques. Moreover, it scales better in the number of nodes in the pose graph, enabling estimation over large-scale graphs ($> 10^5$ nodes) in few seconds. The quality of the approximation degrades gracefully in more challenging scenarios (e.g., large orientation noise). Surprisingly, in these extreme conditions, LAGO outperforms most of the related techniques which are frequently trapped in local minima.

*Approximations in related work*: The idea of a linear approximation to bootstrap nonlinear optimization is not new in literature. For instance, Dellaert and Stroupe [16] propose a linear initialization for planar SLAM using bearing-only measurements. Konolige *et al.* [34] bootstrap poses by constructing a spanning tree on the graph, using a breadth-first visit. Similar ideas can be found in other fields, since pose graph optimization and SLAM are closely related to *bundle adjustment* [54] in photogrammetry, and *structure from motion* [26] in computer vision. In computer vision, linear methods based on algebraic errors (e.g., [26], [44]) are often employed for bootstrapping nonlinear techniques. Crandall *et al.* [14] propose to optimize first for the orientations and then for the translations, refining the resulting estimate with the Levenberg-Marquardt algorithm. However, the approach in [14] resorts to iterative algorithms for both orientation and translation estimation, and assumes small angles to estimate orientations. Sharp *et al.* [49] decouple orientation and translation estimation and exploit graph theoretical concepts to force global consistency of the poses. Also in this case, however, the approach leads to iterative algorithms to propagate the correction across cycles in the graph. This approach shares several intuitions with the papers [4], [17], [18], proposed in the SLAM community. An averaging algorithm on manifold is proposed in [22] to impose global consistency of motion estimates. Martinec and Pajdla [39] also start from the idea of solving first for the orientations. In this case the attempt to deal with the fully 3D setup requires some approximations (i.e., relaxation of the orthonormality constraint of the involved rotations matrices). In the present paper we deal with a somehow simpler problem, i.e., *planar* pose graph optimization; the planar setup (less common in bundle adjustment problems), however, is of great interest for real robotic applications in which the robot operates in indoor scenarios (e.g., warehouses or domestic scenarios). The advantage of this simpler setup is that it allows a closed-form solution of orientation estimation (first phase of LAGO). Moreover, it leads to a very sparse formulation that can be solved efficiently in practice. To the best of our knowledge the idea of including regularization terms to frame orientation

estimation in a linear estimation framework is not reported in related work.

*Previous work from the authors*: The present paper extends previous works [8], [9], [12]; in particular we improved the mentioned articles by including: (i) an analytical assessment of the suboptimality gap of the approach, (ii) several experimental results with an evaluation of accuracy and computational effort of LAGO, and (iii) comparisons with state-of-the-art approaches, including Toro, g2o, and iSAM. Finally, in appendix, we show that some steps of the proposed algorithm involve a complexity that depends on the number of loop-closings in the graph, rather than on the number of poses.

Datasets, source code of LAGO, and the technical report [10], containing extended proofs and other accessory results, are available at [7].

**Notation.** $\mathbf{I}_n$ denotes the $n \times n$ identity matrix, $\mathbf{0}_n$ denotes a (column) vector of all zeros of dimension $n$. $\boldsymbol{M}_{n\times m}$ denotes a matrix with $n$ rows and $m$ columns and $\otimes$ denotes the Kronecker product. The cardinality of a set $S$ is written as $|S|$. Vectors and matrices are in bold face. $\|\boldsymbol{M}\|$ denotes the spectral norm (maximum singular value) of a matrix $\boldsymbol{M}$, or the standard Euclidean norm, for vectors. The notation $\|\boldsymbol{v}\|^2_{\boldsymbol{M}} = \boldsymbol{v}^{\mathsf{T}}\boldsymbol{M}^{-1}\boldsymbol{v}$ denotes the squared Mahalanobis distance.

A *directed* graph $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of elements, called *vertices* or *nodes*, and $\mathcal{E}$ is a set containing unordered pairs of nodes. A generic element $e \in \mathcal{E}$, referred to as *edge*, is in the form $e = (i, j)$, meaning that edge $e$, incident on nodes $i$ and $j$, leaves $i$ (*tail*) and is directed towards node $j$ (*head*) [32]. The number of nodes and edges are denoted with $n + 1$ (this choice will simplify the notation later on) and $m$, respectively, i.e., $|\mathcal{V}| = n+1$ and $|\mathcal{E}| = m$. The *incidence matrix* $\overline{\boldsymbol{A}}$ of a directed graph is a matrix in $\mathbb{Z}^{(n+1)\times m}$ in which each column contains the information of an edge in $\mathcal{E}$; in particular the column corresponding to the edge $e = (i, j)$, has the $i$-th element equal to $-1$, the $j$-th element equal to $+1$ and all the others equal to zero. A directed graph is *connected* (or *weakly connected*) if there exists an undirected path (regardless edges orientation) connecting any pair of nodes. A *spanning tree* of $\mathcal{G}$ is a subgraph with $n$ edges that contains all nodes in $\mathcal{G}$. The edges of $\mathcal{G}$ that do not belong to a given spanning tree of the graph are called *chords* of the graph [13].

## II. Problem Formulation

Let $\boldsymbol{x} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_n\}$ be a set of $n + 1$ poses, describing position and orientation of a mobile robot at consecutive time instants. In a planar setup $\boldsymbol{x}_i = [\boldsymbol{p}_i^{\mathsf{T}}\ \theta_i]^{\mathsf{T}} \in \mathrm{SE(2)}$, where $\boldsymbol{p}_i \in \mathbb{R}^2$ is the Cartesian position of the $i$-th pose, and $\theta_i \in \mathrm{SO(2)}$ is the corresponding orientation. We shall call $\boldsymbol{x}$ a *configuration* of poses. Suppose that it is possible to measure the relative pose between some pairs of poses, say $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:

$$\bar{\boldsymbol{\xi}}_{ij} = \boldsymbol{x}_j \ominus \boldsymbol{x}_i \doteq \begin{bmatrix} \boldsymbol{R}_i^{\mathsf{T}}(\boldsymbol{p}_j - \boldsymbol{p}_i) \\ \theta_j - \theta_i \end{bmatrix}, \tag{1}$$

where $\ominus$ is the *pose compounding operator* (see equation (21)-(24) in [38]) and $\boldsymbol{R}_i$ is a planar rotation matrix of an angle $\theta_i$. Since relative pose measurements are affected by noise, the measured quantities are in the form $\boldsymbol{\xi}_{ij} = \bar{\boldsymbol{\xi}}_{ij} + \boldsymbol{\epsilon}_{ij}$[1], where $\boldsymbol{\epsilon}_{ij} \in \mathbb{R}^3$ is a zero mean Gaussian noise, i.e., $\boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\mathbf{0}_3, \boldsymbol{P}_{ij})$, being $\boldsymbol{P}_{ij}$ a 3 by 3 covariance matrix. In practice, the available relative pose measurements can be classified as:

- *odometric constraints*: relative measurements between poses assumed by the robot at consecutive instants of time. These constraints are connected to measurements of the ego-motion (*odometry*) of the robot and are provided by proprioceptive sensors (wheel odometry, IMU, etc.) or by exteroceptive sensors-based techniques (scan matching, visual features registration, etc.);
- *loop closing constraints*: are connected to place revisiting episodes. Once the robot recognizes that the actual observation matches with a past measurement, it can measure the relative pose between the actual location and the pose from which the past observation was made. This phase requires the use of exteroceptive sensors (e.g., vision sensors, laser range finders).

A pictorial representation of the two types of constraints is provided in Figure 1. The pose graph optimization problem can be then enunciated as follows.

**Problem 1** (Pose graph optimization)**.** *Given the relative pose measurement $\boldsymbol{\xi}_{ij}$ and the corresponding covariance matrix $\boldsymbol{P}_{ij}$, for all available pairs $(i, j)$, estimate the configuration $\boldsymbol{x}^{\mathrm{ML}}$ that maximizes measurement likelihood.*

In SLAM literature it is common to distinguish the determination of the inter-nodal constraints (SLAM *front-end*), from the estimation of robot poses given the constraints (SLAM *back-end*). In this context, we assume the constraints to be given (i.e., we focus on the SLAM back-end), since the reliable determination of both odometric and loop closing constraints is still an active research topic [5], [48], whose implications are out of the scope of the present article.

The pose estimation problem can be naturally modeled using graph formalism: let us consider a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (the *pose graph*) with $n + 1$ nodes, and where the edge set $\mathcal{E}$ contains the node pairs $(i, j)$ such that a relative pose measurement exists between node $i$ and $j$. By convention, if an edge is directed from node $i$ to node $j$, the corresponding relative measurement is expressed in the reference frame of node $i$. The cardinality of the set $\mathcal{E}$ is $m$ (number of available measurements). Therefore, the objective is to associate an absolute pose to each node in the graph. In topological graph theory this problem is also referred to as graph *embedding*, *realization*, or *drawing*, depending on the context and on problem constraints [25].

Since absolute poses are not observable given only relative measurements, according to standard procedure, we set the initial pose of the robot to be the origin of the reference frame, i.e., $\boldsymbol{x}_0 = \mathbf{0}_3$. Therefore, our objective is to estimate a configuration $\boldsymbol{x}^{\mathrm{ML}} = \{\boldsymbol{x}_1^{\mathrm{ML}}, \ldots, \boldsymbol{x}_n^{\mathrm{ML}}\}$, that maximizes the likelihood of the observations, where we excluded the pose $\boldsymbol{x}_0$ from the to-be-computed configuration $\boldsymbol{x}^{\mathrm{ML}}$. Since the

[1] A more formal definition of the uncertainty is $\boldsymbol{\epsilon}_{ij} = \phi^{-1}(\boldsymbol{\xi}_{ij} \ominus \bar{\boldsymbol{\xi}}_{ij})$, where $\phi$ is the *exponential map* from a vector of the Lie algebra to an element of the Lie group. For sake of clarity, we use a less formal definition, with the understanding that the manifold locally behaves as a vector space.

involved densities are Gaussians, it is easy to demonstrate that maximizing measurement likelihood is equivalent to minimizing the sum of the weighted residual errors $\boldsymbol{r}_{ij}(\boldsymbol{x})$ [43]:

$$f(\boldsymbol{x}) \doteq \sum_{(i,j)\in\mathcal{E}} \|(\boldsymbol{x}_j \ominus \boldsymbol{x}_i - \boldsymbol{\xi}_{ij})\|^2_{\boldsymbol{P}_{ij}} \doteq \sum_{(i,j)\in\mathcal{E}} \|\boldsymbol{r}_{ij}(\boldsymbol{x})\|^2_{\boldsymbol{P}_{ij}} . \tag{2}$$

Ordering the available measurements from 1 to $m$, we define the *measurement vector* $\boldsymbol{\xi} = [\boldsymbol{\xi}_1^\mathsf{T} \ \boldsymbol{\xi}_2^\mathsf{T} \ \dots \ \boldsymbol{\xi}_m^\mathsf{T}]^\mathsf{T}$ and the corresponding covariance matrix $\boldsymbol{P}$, that is a block diagonal matrix in which the $k$-th 3 by 3 diagonal block contains the covariance matrix of the $k$-th relative pose measurement. Similarly, we can build the *residual error vector* $\boldsymbol{r}(\boldsymbol{x}) = [\boldsymbol{r}_1^\mathsf{T}(\boldsymbol{x}) \ \boldsymbol{r}_2^\mathsf{T}(\boldsymbol{x}) \ \dots \ \boldsymbol{r}_m^\mathsf{T}(\boldsymbol{x})]^\mathsf{T}$, including the residual errors $\boldsymbol{r}_{ij}(\boldsymbol{x})$ for each available measurement. According to the previous definitions we can write (2) in compact form as:

$$f(\boldsymbol{x}) = \|\boldsymbol{r}(\boldsymbol{x})\|^2_{\boldsymbol{P}} . \tag{3}$$

The batch SLAM problem is hence formulated as a minimization of the non-convex cost function (3), i.e., the maximum likelihood estimate of robot poses is:

$$\boldsymbol{x}^{\mathrm{ML}} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}). \tag{4}$$

The non-convexity of the function $f(\boldsymbol{x})$ is due to the structure of the pose compounding operator (1) and, in particular, to the nonlinear terms in the orientations of the robot. State-of-the-art approaches use nonlinear optimization techniques for solving (4), as briefly recalled in the following.

*Iterative Nonlinear Optimization*: State-of-the-art iterative approaches to pose graph optimization refine a given initial guess by repeatedly optimizing local convex approximations of the cost function (3). Let $\boldsymbol{x}^{(\tau)}$ be the configuration computed by the algorithm at iteration $\tau$, being $\boldsymbol{x}^{(0)}$ the given initial guess. At each iteration, nonlinear techniques linearize $\boldsymbol{r}(\boldsymbol{x})$ around the current solution $\boldsymbol{x}^{(\tau)}$, obtaining

$$\boldsymbol{r}(\boldsymbol{x}) \simeq \boldsymbol{r}(\boldsymbol{x}^{(\tau)}) + \boldsymbol{J}_{\boldsymbol{r}} \, \boldsymbol{\eta}_{\boldsymbol{x}}^{(\tau)}, \tag{5}$$

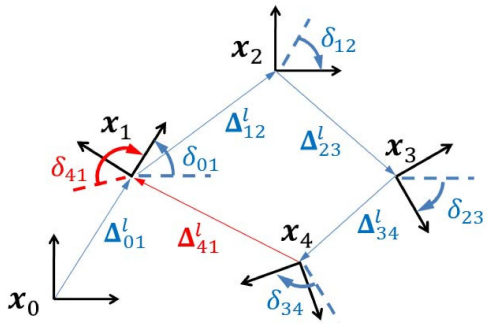
where $\boldsymbol{\eta}_{\boldsymbol{x}}^{(\tau)} \in \mathbb{R}^{3n}$ is the displacement from the linearization point and $\boldsymbol{J}_{\boldsymbol{r}} = \boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{x}^{(\tau)}) \in \mathbb{R}^{3m\times 3n}$ is the Jacobian of the residual error function $\boldsymbol{r}(\cdot)$. Therefore, related work approximates the cost function (3) around $\boldsymbol{x}^{(\tau)}$ with the convex objective:

$$f \simeq \left\| \boldsymbol{r}(\boldsymbol{x}^{(\tau)}) + \boldsymbol{J}_{\boldsymbol{r}} \, \boldsymbol{\eta}_{\boldsymbol{x}}^{(\tau)} \right\|^2_{\boldsymbol{P}} \tag{6}$$

The previous cost function is quadratic in the unknown $\boldsymbol{\eta}_{\boldsymbol{x}}^{(\tau)}$ and its minimizer corresponds to the solution of the *normal equation* [43]:

$$\left(\boldsymbol{J}_{\boldsymbol{r}}^\mathsf{T} \boldsymbol{P}^{-1} \boldsymbol{J}_{\boldsymbol{r}}\right) \boldsymbol{\eta}_{\boldsymbol{x}}^{(\tau)} = -\boldsymbol{J}_{\boldsymbol{r}}^\mathsf{T} \, \boldsymbol{P}^{-1} \, \boldsymbol{r}(\boldsymbol{x}^{(\tau)}). \tag{7}$$

Therefore, at each iteration the configuration is refined using the local correction $\boldsymbol{\eta}_{\boldsymbol{x}}^{(\tau)}$ and the process is repeated until convergence. In iterative approaches the source of complexity stems from the need of repeatedly solving the large-scale linear system (7). Related works address this issue employing *direct* methods (usually based on QR or Cholesky factorization) or *iterative* methods (e.g., Gauss-Seidel relaxation, multi-level relaxation). A discussion on the performances of *direct* and *iterative* methods for solving linear systems, with application to SLAM, can be found in [15].




$$\overline{\boldsymbol{A}} = \underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ +1 & -1 & 0 & 0 & +1 \\ 0 & +1 & -1 & 0 & 0 \\ 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & 0 & +1 & -1 \end{bmatrix}}_{\boldsymbol{A}} \quad \boldsymbol{A}^\mathsf{T} = \begin{bmatrix} +1 & 0 & 0 & 0 \\ -1 & +1 & 0 & 0 \\ 0 & -1 & +1 & 0 \\ 0 & 0 & -1 & +1 \\ +1 & 0 & 0 & -1 \end{bmatrix}$$

Fig. 1. Pose graph with 5 nodes: a reference frame is attached to each node to define the corresponding pose; odometric constraints are shown in blue, whereas an example of loop closing constraint is drawn in red. The figure also shows the relative position measurements $\{\boldsymbol{\Delta}^l_{01}, \boldsymbol{\Delta}^l_{12}, \boldsymbol{\Delta}^l_{23}, \boldsymbol{\Delta}^l_{34}, \boldsymbol{\Delta}^l_{41}\}$, the relative orientation measurements $\{\delta_{01}, \delta_{12}, \delta_{23}, \delta_{34}, \delta_{41}\}$, the incidence matrix $\overline{\boldsymbol{A}}$, and the reduced incidence matrix $\boldsymbol{A}$.

## III. On the Structure of Pose Graph Optimization

We first observe that each relative pose measurement includes a vector in $\mathbb{R}^2$ (*relative position*) corresponding to the first two components of $\boldsymbol{\xi}_{ij}$, in (1), and a scalar (*relative orientation*), see also the toy example in Figure 1. Therefore we can rewrite each measurement as $\boldsymbol{\xi}_{ij} = [(\boldsymbol{\Delta}^l_{ij})^\mathsf{T} \ \delta_{ij}]^\mathsf{T}$, where the superscript $l$ remarks that the relative position vector is expressed in a local frame. According to the notation introduced so far the cost function (2) can be rewritten as:

$$f(\boldsymbol{x}) = \sum_{(i,j)\in\mathcal{E}} \left\| \begin{bmatrix} \boldsymbol{R}_i^\mathsf{T}(\boldsymbol{p}_j - \boldsymbol{p}_i) - \boldsymbol{\Delta}^l_{ij} \\ (\theta_j - \theta_i) - \delta_{ij} \end{bmatrix} \right\|^2_{\boldsymbol{P}_{ij}} . \tag{8}$$

In this context we assume that the relative position measurements and the relative orientation measurements are independent, i.e. $\boldsymbol{P}_{ij} = \mathrm{diag}(\boldsymbol{P}_{\boldsymbol{\Delta}^l_{ij}}, \boldsymbol{P}_{\delta_{ij}})$. Under this hypothesis the cost function $f(\boldsymbol{x})$ becomes:

$$\begin{aligned} f(\boldsymbol{x}) = &\sum_{(i,j)\in\mathcal{E}} \left\|\boldsymbol{R}_i^\mathsf{T}(\boldsymbol{p}_j - \boldsymbol{p}_i) - \boldsymbol{\Delta}^l_{ij}\right\|^2_{\boldsymbol{P}_{\boldsymbol{\Delta}^l_{ij}}} + \\ &+ \sum_{(i,j)\in\mathcal{E}} \|(\theta_j - \theta_i) - \delta_{ij}\|^2_{\boldsymbol{P}_{\delta_{ij}}} . \end{aligned} \tag{9}$$

In order to put the previous formulation in a more compact form, we number the available measurements from 1 to $m$; we stack all the relative position measurements in the vector $\boldsymbol{\Delta}^l = [(\boldsymbol{\Delta}^l_1)^\mathsf{T} \ (\boldsymbol{\Delta}^l_2)^\mathsf{T} \ \dots \ (\boldsymbol{\Delta}^l_m)^\mathsf{T}]^\mathsf{T}$, and all the relative orientation measurements in the vector $\boldsymbol{\delta} = [\delta_1 \ \delta_2 \ \dots \ \delta_m]^\mathsf{T}$. Repeating the same procedure for the positions and orientations assumed

by the robot we get the *nodes position* $\boldsymbol{p} = [\boldsymbol{p}_1^\mathsf{T} \ \dots \ \boldsymbol{p}_n^\mathsf{T}]^\mathsf{T}$ and the *nodes orientation* $\boldsymbol{\theta} = [\theta_1 \ \dots \ \theta_n]^\mathsf{T}$. Notice that we excluded $\boldsymbol{p}_0$ from $\boldsymbol{p}$ and $\theta_0$ from $\boldsymbol{\theta}$, as the first node was set to the origin of the reference frame (hence it is not involved in the optimization), i.e., $\boldsymbol{p}_0 = \mathbf{0}_2$ and $\theta_0 = 0$. Therefore, equation (9) can be rewritten as:

$$f(\boldsymbol{x}) = \left\| \boldsymbol{A}_2^\mathsf{T}\boldsymbol{p} - \boldsymbol{R}\boldsymbol{\Delta}^l \right\|^2_{\left(\boldsymbol{R}\boldsymbol{P}_{\boldsymbol{\Delta}^l}\boldsymbol{R}^\mathsf{T}\right)} + \left\| \boldsymbol{A}^\mathsf{T}\boldsymbol{\theta} - \boldsymbol{\delta} \right\|^2_{\boldsymbol{P}_\delta}, \tag{10}$$

where $\boldsymbol{A}$ is the *reduced incidence matrix* [2] of the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (an intuitive explanation of the structure of $\boldsymbol{A}^\mathsf{T}$ is given in Figure 1), $\boldsymbol{A}_2 = \boldsymbol{A} \otimes \mathbf{I}_2$ is an expanded form of the incidence matrix [2], $\boldsymbol{P}_{\boldsymbol{\Delta}^l} = \text{diag}(\boldsymbol{P}_{\boldsymbol{\Delta}^l_1}, \boldsymbol{P}_{\boldsymbol{\Delta}^l_2}, \dots, \boldsymbol{P}_{\boldsymbol{\Delta}^l_m})$, $\boldsymbol{P}_\delta = \text{diag}(\boldsymbol{P}_{\delta_1}, \boldsymbol{P}_{\delta_2}, \dots, \boldsymbol{P}_{\delta_m})$, and $\boldsymbol{R} = \boldsymbol{R}(\boldsymbol{\theta})$ is a block diagonal matrix containing the rotation matrices that transform the corresponding local measurements in the global frame, i.e., the non-zero entries of $\boldsymbol{R}$ are in positions $(2k-1, 2k-1), (2k-1, 2k), (2k, 2k-1), (2k, 2k),\ k = 1, \dots, m$, and the $k$-th diagonal block contains the rotation matrix converting the $k$-th relative position measurement in the global frame. The expanded matrix $\boldsymbol{A}_2$, or the more general form $\boldsymbol{A}_u = \boldsymbol{A} \otimes \mathbf{I}_u$, has the same structure of $\boldsymbol{A}$, but the zero elements in $\boldsymbol{A}$ are replaced with null matrices $\mathbf{0}_{u\times u}$, whereas the elements of value $+1$ and $-1$ are replaced by $\mathbf{I}_u$ and $-\mathbf{I}_u$, respectively.

We observe that the minimization of the cost function (10) is, by definition, equivalent to find the solution (in the least-squares sense) to the following system of equations:

$$\begin{cases} \boldsymbol{A}_2^\mathsf{T}\boldsymbol{p} = \boldsymbol{R}(\boldsymbol{\theta})\boldsymbol{\Delta}^l \\ \boldsymbol{A}^\mathsf{T}\boldsymbol{\theta} = \boldsymbol{\delta} \end{cases}. \tag{11}$$

Roughly speaking, since system (11) is overdetermined and the input data are noisy, no configuration (in general) will satisfy exactly all the constraints, and a (least squares) solution of the system is the one minimizing the weighted residual errors; by definition, a global minimizer of the cost function (10) attains the minimum of the weighted residual errors. The nonlinear nature of the problem is connected with the matrix $\boldsymbol{R}(\boldsymbol{\theta})$, which contains trigonometric functions of nodes orientations.

*Estimating orientations from relative orientation measurements*: Observing (11) one may notice that the second subset of equations ($\boldsymbol{A}^\mathsf{T}\boldsymbol{\theta} = \boldsymbol{\delta}$) is linear in the unknown $\boldsymbol{\theta}$. Therefore, the computation of $\boldsymbol{\theta}$ from the sole relative orientation measurements $\boldsymbol{\delta}$ is a linear estimation problem and can be solved in closed-form. We will exploit this first observation in Section IV-A. We anticipate that the actual orientation measurements are not really linear: we are assuming that the relative orientation measurements are $\delta_{ij} = \theta_j - \theta_i$, but the actual relative orientations are only measured modulo $2\pi$, since the angles belong to the manifold $\text{SO}(2)$. We will come back to this point in Section V, where we will show how to reformulate the measurement model in a linear form.

*Pose graph optimization with known robot orientations*: Before presenting our closed-form approximation, we observe that, when the orientations of the robot are known, the pose estimation problem becomes linear. In fact, under this assumption, the second equation in problem (11) disappears and the pose estimation reduces to *positioning* (only $\boldsymbol{p}$ has to be estimated). If $\boldsymbol{\theta}$ is equal to a known $\boldsymbol{\theta}^\star$, we can rewrite (11) as:

$$\boldsymbol{A}_2^\mathsf{T}\boldsymbol{p} = \boldsymbol{R}^\star\boldsymbol{\Delta}^l, \tag{12}$$

with $\boldsymbol{R}^\star = \boldsymbol{R}(\boldsymbol{\theta}^\star)$. If we define $\boldsymbol{\Delta}^g = \boldsymbol{R}^\star\boldsymbol{\Delta}^l$ and $\boldsymbol{P}_{\boldsymbol{\Delta}^g} = (\boldsymbol{R}^\star)\boldsymbol{P}_{\boldsymbol{\Delta}^l}(\boldsymbol{R}^\star)^\mathsf{T}$, the maximum likelihood estimate for the linear estimation problem (12) can be computed in closed form:

$$\begin{aligned} \boldsymbol{p}^\star &= \arg\min_{\boldsymbol{p}} \left\| \boldsymbol{A}_2^\mathsf{T}\boldsymbol{p} - \boldsymbol{\Delta}^g \right\|^2_{\boldsymbol{P}_{\boldsymbol{\Delta}^g}} \\ \implies \boldsymbol{p}^\star &= (\boldsymbol{A}_2\boldsymbol{P}^{-1}_{\boldsymbol{\Delta}^g}\boldsymbol{A}_2^\mathsf{T})^{-1}\boldsymbol{A}_2\boldsymbol{P}^{-1}_{\boldsymbol{\Delta}^g}\boldsymbol{\Delta}^g. \end{aligned} \tag{13}$$

The linear estimator $\boldsymbol{p}^\star$ is itself a random vector, being function of the measurements, and can be proven to be Normally distributed with covariance matrix $(\boldsymbol{A}_2\boldsymbol{P}^{-1}_{\boldsymbol{\Delta}^g}\boldsymbol{A}_2^\mathsf{T})^{-1}$ [40]. We notice that $\boldsymbol{\Delta}^g$ and $\boldsymbol{P}_{\boldsymbol{\Delta}^g}$ are respectively the relative position measurements and the measurement covariance, expressed in the global frame; therefore, the linearity of this simplified problem stems from the capability of expressing the relative position measurements in a global frame. Such setup has been extensively investigated in both centralized and distributed setups, with application to multi agent localization [1], [2]; in practice, it is an effective solution to SLAM (in both 2D and 3D problems) for the cases in which a reliable absolute orientation information is available.

In the next section, we extend this simple linear case to the more general setup in which nodes orientation is unknown. Before presenting the contribution we state a basic result pertinent to the discussion in this paper. The proof is omitted for space reasons and can be found in [10].

**Lemma 1** (Properties of the pose graph)**.** *The following properties hold true for a pose graph modeling the batch SLAM problem:*

1) *The pose graph $\mathcal{G}$, including odometric and loop closing constraints, is connected;*
2) *The edges corresponding to odometric constraints in the pose graph constitute a spanning tree for $\mathcal{G}$;*
3) *If $\mathsf{T}$ is the spanning tree comprising all odometric constraints, the edges corresponding to loop closing constraints are chords of $\mathcal{G}$, with respect to $\mathsf{T}$;*
4) *The reduced incidence matrix $\boldsymbol{A}$ of graph $\mathcal{G}$, and its augmented version $\boldsymbol{A}_2$ have full column rank.* □

## IV. An Approximation for Pose Graph Optimization

In the previous section we noticed that the inference over robot poses includes a linear estimation subproblem (estimate robot orientations from relative orientation measurements); moreover, we observed that, given robot orientations, we can express the relative position measurements in a global frame, making position estimation a linear estimation problem. In this section we exploit these observations to propose a linear approximation for pose graph optimization. The approximation, named LAGO (*Linear Approximation for pose Graph Optimization*), is structured in three phases. Sections IV-A to IV-C detail each phase of the algorithm. Section IV-D provides an assessment of the approximation; in particular, (i) we guarantee the existence of the approximation (Proposition 3), (ii) we show that the approximation corresponds to a

Gauss-Newton step computed around a suboptimal solution $\hat{\boldsymbol{x}}$ (Proposition 4), and (iii) we provide an analytical assessment of how far is $\hat{\boldsymbol{x}}$ from optimality (Proposition 5). Finally, in Section IV-E we provide a probabilistic interpretation of the approach.

Before entering in the details of each phase of LAGO we provide an algorithmic overview of the approach, that can help the reader to keep the overall vision, while proceeding through Sections IV-A to IV-C.

*Overview*: The proposed *Linear Approximation for pose Graph Optimization* (LAGO) can be computed using the following algorithm, structured in three phases:

1) Solve the linear orientation estimation subproblem:
$$\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\theta} = \boldsymbol{\delta}, \tag{14}$$
from which the suboptimal orientation estimate $\hat{\boldsymbol{\theta}}$ and its covariance matrix $\boldsymbol{P}_{\hat{\boldsymbol{\theta}}}$ can be obtained.
2) Express the relative position measurements in the global reference frame using the orientation estimate $\hat{\boldsymbol{\theta}}$:
$$\boldsymbol{z} = \begin{bmatrix} \hat{\boldsymbol{R}}\boldsymbol{\Delta}^l \\ \hat{\boldsymbol{\theta}} \end{bmatrix} \doteq \begin{bmatrix} g_1(\boldsymbol{\Delta}^l, \boldsymbol{\theta}) \\ g_2(\boldsymbol{\theta}) \end{bmatrix}_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}, \tag{15}$$
with $\hat{\boldsymbol{R}} = \boldsymbol{R}(\hat{\boldsymbol{\theta}})$; compute the corresponding uncertainty (preserving the correlation with the orientation estimate):
$$\boldsymbol{P}_{\boldsymbol{z}} = \boldsymbol{H} \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{\Delta}^l} & \boldsymbol{0}_{2m\times n} \\ \boldsymbol{0}_{2m\times n}^{\mathsf{T}} & \boldsymbol{P}_{\hat{\boldsymbol{\theta}}} \end{bmatrix} \boldsymbol{H}^{\mathsf{T}}, \tag{16}$$
where $\boldsymbol{H}$ is the Jacobian of the transformation in (15):
$$\boldsymbol{H} \doteq \begin{bmatrix} \frac{\partial g_1}{\partial \boldsymbol{\Delta}^l} & \frac{\partial g_1}{\partial \boldsymbol{\theta}} \\ \frac{\partial g_2}{\partial \boldsymbol{\Delta}^l} & \frac{\partial g_2}{\partial \boldsymbol{\theta}} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{R}} & \boldsymbol{J} \\ \boldsymbol{0}_{2m\times n}^{\mathsf{T}} & \mathbf{I}_n \end{bmatrix}. \tag{17}$$
3) Solve the following linear estimation problem in the unknown $\boldsymbol{x} = [\boldsymbol{p}^{\mathsf{T}}\ \boldsymbol{\theta}^{\mathsf{T}}]^{\mathsf{T}}$, given $\boldsymbol{z}$, (15), and $\boldsymbol{P}_{\boldsymbol{z}}$, (16):
$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{A}_2^{\mathsf{T}} & \boldsymbol{0}_{2m\times n} \\ \boldsymbol{0}_{n\times 2n} & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\theta} \end{bmatrix} \doteq \boldsymbol{B}^{\mathsf{T}}\boldsymbol{x}, \tag{18}$$
from which it is possible to compute the linear approximation $\boldsymbol{x}^\star = [(\boldsymbol{p}^\star)^{\mathsf{T}}\ (\boldsymbol{\theta}^\star)^{\mathsf{T}}]^{\mathsf{T}}$.

### A. Phase 1: Linear Orientation Estimation

The first phase of LAGO requires the solution of the second subset of constraints of (11); this is a standard linear estimation problem: given the matrix $\boldsymbol{A}$, the measurements $\boldsymbol{\delta}$ and the corresponding covariance matrix $\boldsymbol{P}_{\boldsymbol{\delta}}$, the objective is to provide an estimate $\hat{\boldsymbol{\theta}}$ of the unknown $\boldsymbol{\theta}$. According to well known results in linear estimation theory [40], the maximum likelihood estimate of $\boldsymbol{\theta}$ in (14) is
$$\hat{\boldsymbol{\theta}} = (\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}\boldsymbol{\delta}, \tag{19}$$
and the corresponding covariance is:
$$\boldsymbol{P}_{\hat{\boldsymbol{\theta}}} = (\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}\boldsymbol{A}^{\mathsf{T}})^{-1}. \tag{20}$$

We already observed that, once the absolute orientation of the robot is known, also the first equation in (11) becomes linear in the unknown $\boldsymbol{p}$. Therefore, using $\hat{\boldsymbol{\theta}}$ as the true nodes orientation, we could also compute an estimate for nodes position $\hat{\boldsymbol{p}}$, through linear estimation:
$$\hat{\boldsymbol{p}} = \left[\boldsymbol{A}_2 \left(\hat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{\Delta}^l}\hat{\boldsymbol{R}}^{\mathsf{T}}\right)^{-1}\boldsymbol{A}_2^{\mathsf{T}}\right]^{-1}\boldsymbol{A}_2 \left(\hat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{\Delta}^l}\hat{\boldsymbol{R}}^{\mathsf{T}}\right)^{-1}\hat{\boldsymbol{R}}\boldsymbol{\Delta}^l, \tag{21}$$
where $\hat{\boldsymbol{R}} = \boldsymbol{R}(\hat{\boldsymbol{\theta}})$. However, the first equation in (11) also constraints the orientations of the robot, thus the estimate $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{p}}^{\mathsf{T}}\ \hat{\boldsymbol{\theta}}^{\mathsf{T}}]^{\mathsf{T}}$ constitutes a *suboptimal solution*, in which the influence of the first equation on the estimated orientations is neglected. Such approach corresponds to optimize the second summand in the cost function (10) with respect to $\boldsymbol{\theta}$, to substitute this suboptimal solution in the first summand, and to optimize it with respect to the position variables $\boldsymbol{p}$. The obtained solution, however, does not correspond to a minimum of the overall cost function.

### B. Phase 2: First-order Error Propagation

According to linear estimation theory, the estimator $\hat{\boldsymbol{\theta}}$ given in (19) is a Normally distributed random variable with covariance $\boldsymbol{P}_{\hat{\boldsymbol{\theta}}}$, as per (20). The second phase of LAGO exploits this fact to express the relative position measurements in the global reference frame: the vector $\boldsymbol{z}$ is in the form $\boldsymbol{z} = [(\boldsymbol{\Delta}^g)^{\mathsf{T}}\ \hat{\boldsymbol{\theta}}^{\mathsf{T}}]^{\mathsf{T}}$ where $\boldsymbol{\Delta}^g = \hat{\boldsymbol{R}}\boldsymbol{\Delta}^l$ is an estimate of the relative nodes positions expressed in the global frame. The covariance matrix of $\boldsymbol{z}$ can be obtained by a first-order propagation of the uncertainty, according to (16); for the following derivation, it is useful to compute the products in (16) obtaining
$$\boldsymbol{P}_{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{\Delta}^g} + \boldsymbol{J}\boldsymbol{P}_{\hat{\boldsymbol{\theta}}}\boldsymbol{J}^{\mathsf{T}} & \boldsymbol{J}\boldsymbol{P}_{\hat{\boldsymbol{\theta}}} \\ \boldsymbol{P}_{\hat{\boldsymbol{\theta}}}\boldsymbol{J}^{\mathsf{T}} & \boldsymbol{P}_{\hat{\boldsymbol{\theta}}} \end{bmatrix}, \tag{22}$$
where $\boldsymbol{P}_{\boldsymbol{\Delta}^g} \doteq \hat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{\Delta}^l}\hat{\boldsymbol{R}}^{\mathsf{T}}$, while $\boldsymbol{J}$ and $\boldsymbol{P}_{\hat{\boldsymbol{\theta}}}$ are defined as in (17) and (20).

The trick of including the orientation estimates in $\boldsymbol{z}$ is useful to preserve the correlation between the relative position measurements (expressed in the global frame) and the corresponding angular information. As we will see in the following section, such correlation terms play a fundamental role in the outcome of LAGO.

### C. Phase 3: Linear Pose Estimation

We now want to show that the last phase allows to correct the suboptimal configuration estimate $\hat{\boldsymbol{x}} \doteq [\hat{\boldsymbol{p}}^{\mathsf{T}}\ \hat{\boldsymbol{\theta}}^{\mathsf{T}}]^{\mathsf{T}}$, i.e.
$$\boldsymbol{\theta}^\star = \hat{\boldsymbol{\theta}} + \tilde{\boldsymbol{\theta}}, \qquad \boldsymbol{p}^\star = \hat{\boldsymbol{p}} + \tilde{\boldsymbol{p}}, \tag{23}$$
in which $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{p}}$ are computed as in equations (19) and (21), while $\tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{p}}$ are local correction terms. The nature of the correction terms $\tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{p}}$ is discussed in Section IV-D, while we now show that the outcome of LAGO in the form (23). In order to proceed in the demonstration we compute $\boldsymbol{x}^\star = [(\boldsymbol{p}^\star)^{\mathsf{T}}\ (\boldsymbol{\theta}^\star)^{\mathsf{T}}]^{\mathsf{T}}$ from the linear estimation problem (18). As usual, we use linear estimation theory to obtain the estimate:
$$\boldsymbol{x}^\star = \begin{bmatrix} \boldsymbol{p}^\star \\ \boldsymbol{\theta}^\star \end{bmatrix} = (\boldsymbol{B}\boldsymbol{P}_{\boldsymbol{z}}^{-1}\boldsymbol{B}^{\mathsf{T}})^{-1}\boldsymbol{B}\boldsymbol{P}_{\boldsymbol{z}}^{-1}\boldsymbol{z}. \tag{24}$$

In order to write $\boldsymbol{p}^\star$ and $\boldsymbol{\theta}^\star$ in explicit form, we compute $\boldsymbol{P}_{\boldsymbol{z}}^{-1}$ using standard blockwise matrix inversion rules [27]:

$$\boldsymbol{P}_{\boldsymbol{z}}^{-1} = \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1} & -\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J} \\ -\boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1} & \boldsymbol{P}_{\hat{\boldsymbol{\theta}}}^{-1} + \boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J} \end{bmatrix}. \tag{25}$$

We then compute the matrix $\boldsymbol{\Omega}_{\boldsymbol{x}^\star} \doteq (\boldsymbol{B}\boldsymbol{P}_{\boldsymbol{z}}^{-1}\boldsymbol{B}^\mathsf{T})$:

$$\boldsymbol{\Omega}_{\boldsymbol{x}^\star} = \begin{bmatrix} \boldsymbol{P}_{\hat{\boldsymbol{p}}}^{-1} & -\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J} \\ -\boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{A}_2^\mathsf{T} & \boldsymbol{P}_{\hat{\boldsymbol{\theta}}}^{-1} + \boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J} \end{bmatrix}, \tag{26}$$

where $\boldsymbol{P}_{\hat{\boldsymbol{p}}} \doteq (\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{A}_2^\mathsf{T})^{-1}$. The inverse of $\boldsymbol{\Omega}_{\boldsymbol{x}^\star}$, namely $\boldsymbol{P}_{\boldsymbol{x}^\star}$, is in the form:

$$\boldsymbol{P}_{\boldsymbol{x}^\star} = \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{p}^\star} & \boldsymbol{P}_{\boldsymbol{p}^\star,\boldsymbol{\theta}^\star} \\ \boldsymbol{P}_{\boldsymbol{p}^\star,\boldsymbol{\theta}^\star}^\mathsf{T} & \boldsymbol{P}_{\boldsymbol{\theta}^\star} \end{bmatrix}, \tag{27}$$

where:

$$\begin{aligned} \boldsymbol{P}_{\boldsymbol{\theta}^\star} &\doteq \left(\boldsymbol{P}_{\hat{\boldsymbol{\theta}}}^{-1} + \boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J} - \boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{A}_2^\mathsf{T}\boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J}\right)^{-1} \\ \boldsymbol{P}_{\boldsymbol{p}^\star} &= \boldsymbol{P}_{\hat{\boldsymbol{p}}} + \boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J}\boldsymbol{P}_{\boldsymbol{\theta}^\star}\boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{A}_2^\mathsf{T}\boldsymbol{P}_{\hat{\boldsymbol{p}}} \\ \boldsymbol{P}_{\boldsymbol{p}^\star,\boldsymbol{\theta}^\star} &= \boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J}\boldsymbol{P}_{\boldsymbol{\theta}^\star}. \end{aligned} \tag{28}$$

From the expression of $\boldsymbol{P}_{\boldsymbol{x}^\star}$ and $\boldsymbol{P}_{\boldsymbol{z}}^{-1}$ it is straightforward to compute the estimate (24) as:

$$\begin{aligned} \boldsymbol{\theta}^\star \;&=\; (\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}\boldsymbol{A}^\mathsf{T})^{-1}\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}\boldsymbol{\delta} + \\ &+\; \boldsymbol{P}_{\boldsymbol{\theta}^\star}\boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\left(\boldsymbol{A}_2^\mathsf{T}\boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1} - \mathbf{I}_{2m}\right)\hat{\boldsymbol{R}}\boldsymbol{\Delta}^l, \end{aligned} \tag{29}$$

and:

$$\begin{aligned} \boldsymbol{p}^\star \;&=\; \left[\boldsymbol{A}_2\left(\hat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{\Delta}^l}\hat{\boldsymbol{R}}^\mathsf{T}\right)^{-1}\boldsymbol{A}_2^\mathsf{T}\right]^{-1}\boldsymbol{A}_2\left(\hat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{\Delta}^l}\hat{\boldsymbol{R}}^\mathsf{T}\right)^{-1}\hat{\boldsymbol{R}}\boldsymbol{\Delta}^l + \\ &+\; \boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J}\boldsymbol{P}_{\boldsymbol{\theta}^\star}\boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\left(\boldsymbol{A}_2^\mathsf{T}\boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1} - \mathbf{I}_{2m}\right)\hat{\boldsymbol{R}}\boldsymbol{\Delta}^l. \end{aligned} \tag{30}$$

Notice that the previous equation provides a closed-form expression of the outcome of LAGO. In Section VI-A we will point out that there is a more efficient way to compute the proposed approximation, although the explicit solution (29) is useful to assess the quality of the approximation.

Comparing (29) with (19) and (21), we note that $\boldsymbol{x}^\star$ is in the form $\boldsymbol{x}^\star = \hat{\boldsymbol{x}} + \tilde{\boldsymbol{x}}$, where $\tilde{\boldsymbol{x}} = [(\tilde{\boldsymbol{p}})^\mathsf{T} \;\; (\tilde{\boldsymbol{\theta}})^\mathsf{T}]^\mathsf{T}$ and:

$$\begin{aligned} \tilde{\boldsymbol{\theta}} &= \boldsymbol{P}_{\boldsymbol{\theta}^\star}\boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\left(\boldsymbol{A}_2^\mathsf{T}\boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1} - \mathbf{I}_{2m}\right)\hat{\boldsymbol{R}}\boldsymbol{\Delta}^l \\ \tilde{\boldsymbol{p}} &= \boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J}\boldsymbol{P}_{\boldsymbol{\theta}^\star}\boldsymbol{J}^\mathsf{T}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\left(\boldsymbol{A}_2^\mathsf{T}\boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1} - \mathbf{I}_{2m}\right)\hat{\boldsymbol{R}}\boldsymbol{\Delta}^l. \end{aligned} \tag{31}$$

In the following section we show that $\tilde{\boldsymbol{x}}$ is nothing else than the local correction obtained by applying a Gauss-Newton iteration from $\hat{\boldsymbol{x}}$. Moreover, we will quantify how far is $\hat{\boldsymbol{x}}$ from the optimal (maximum likelihood) estimate.

**Remark 2** (Initial guess in iterative optimization techniques)**.** *Standard nonlinear optimization techniques need an initial guess for all the variables in the problem.* LAGO*, instead, exploits problem structure and intrinsically computes an initial guess for orientations (Phase 1 of the approach), while it does not require computing explicitly the guess $\hat{\boldsymbol{p}}$ for nodes positions. The third phase, in fact, directly computes the estimate $\boldsymbol{p}^\star$, which is only a function of the measurements (and the corresponding covariances) and of the estimate $\hat{\boldsymbol{\theta}}$.* □

### *D. Assessment of the Approximation*

In this section we provide three results assessing the quality of the proposed approximation. The first result (Proposition 3) states that the approximation is well posed, in the sense that the linear systems appearing in LAGO always admit a unique solution. The second result (Proposition 4) proves that the correction terms $\tilde{\boldsymbol{x}}$ is equivalent to a Gauss-Newton step from $\hat{\boldsymbol{x}}$. The third result (Proposition 5) provides an upper bound on the suboptimality gap of the estimate $\hat{\boldsymbol{x}}$.

**Proposition 3** (Existence and uniqueness of the approximation)**.** *The approximation computed with* LAGO *is unique if the pose graph $\mathcal{G}$ is connected and the pose of at least one node is supposed to be known.* □

The proof of the claim is in the same spirit of standard results on estimation over graphs [2] and is reported in Appendix A for completeness. Proposition 3 excludes numerical issues due to singularities in the involved matrices, and guarantees that the approximation can be always computed under the usual SLAM setup (connected graph and a pose set to the origin of the reference frame). This is relevant when comparing the proposed technique with related solutions for which the existence is not guaranteed, e.g., [45]. After proving the existence, we now discuss in which sense LAGO approximates the optimal solution of (10).

**Proposition 4** (Linear Approximation)**.** *The approximation computed with* LAGO *is equivalent to a Gauss-Newton iteration from the suboptimal estimate $\hat{\boldsymbol{x}}$. In particular, the third phase produces a local correction of such suboptimal estimate leading it towards a minimum of the cost function* (10)*.* □

The proof of Proposition 4 is provided in Appendix B and uses simple matrix manipulation to show that a Gauss-Newton iteration around $\hat{\boldsymbol{x}}$ produces the same local correction terms obtained with LAGO, as per equation (31). As stressed in Remark 2, $\hat{\boldsymbol{p}}$ is not actually computed by the proposed approach (LAGO only computes $\hat{\boldsymbol{\theta}}$), and it was only introduced in Section IV-A to assess the resulting approximation.

In practice, the suboptimal orientation estimate $\hat{\boldsymbol{\theta}}$ is quite accurate, therefore the third phase is expected to lead the approximation towards a global optimum of problem (10).

As a further contribution, we report in Appendix C an approach for computing $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{p}}$, whose complexity depends on the number of loop closings in the pose graph. In the technical report [10] we also show that the first phase of LAGO can be carried out in incremental fashion, i.e., the estimate $\hat{\boldsymbol{\theta}}$ can be computed by recursively including one relative orientation measurement at a time.

Before proceeding in the experimental evaluation of the approximation, it is interesting to quantify analytically the quality of the suboptimal estimate $\hat{\boldsymbol{x}}$. For sake of simplicity, we will assume a diagonal structure for the covariance matrices describing measurements, i.e., $\boldsymbol{P}_{\boldsymbol{\Delta}^l} = \sigma_\Delta^2\mathbf{I}_{2m}$ and $\boldsymbol{P}_{\boldsymbol{\delta}} = \sigma_\delta^2\mathbf{I}_m$. This assumption will ease the identification of the factors influencing the quality of $\hat{\boldsymbol{x}}$. Under this assumption, the cost function (10) becomes

$$f(\boldsymbol{x}) = \tfrac{1}{\sigma_\Delta^2}\left\|\boldsymbol{A}_2^\mathsf{T}\boldsymbol{p} - \boldsymbol{R}\boldsymbol{\Delta}^l\right\|^2 + \tfrac{1}{\sigma_\delta^2}\left\|\boldsymbol{A}^\mathsf{T}\boldsymbol{\theta} - \boldsymbol{\delta}\right\|^2. \tag{32}$$

Clearly, we can multiply the cost function by a constant without changing the optimal solution; in particular, we multiply $f(\boldsymbol{x})$ by $\sigma_\delta^2$, obtaining the following *normalized cost*:

$$f_n(\boldsymbol{x}) = \frac{\sigma_\delta^2}{\sigma_\Delta^2} \left\| \boldsymbol{A}_2^\mathsf{T} \boldsymbol{p} - \boldsymbol{R}\boldsymbol{\Delta}^l \right\|^2 + \left\| \boldsymbol{A}^\mathsf{T}\boldsymbol{\theta} - \boldsymbol{\delta} \right\|^2. \tag{33}$$

We are now ready to assess the quality of the suboptimal estimate $\hat{\boldsymbol{x}}$.

**Proposition 5** (Suboptimality gap of $\hat{\boldsymbol{x}}$)**.** *Let us define the (normalized) global optimal objective $f_n^{\mathrm{ML}} = f_n(\boldsymbol{x}^{\mathrm{ML}})$, computed at the maximum likelihood estimate $\boldsymbol{x}^{\mathrm{ML}}$; define accordingly, the suboptimal objective $\hat{f}_n = f_n(\hat{\boldsymbol{x}})$, computed at $\hat{\boldsymbol{x}}$. Then, the suboptimality gap $\beta \doteq \hat{f}_n - f_n^{\mathrm{ML}}$ satisfies:*

$$\beta \leq \frac{\sigma_\delta^2}{\sigma_\Delta^2} \|\boldsymbol{\Delta}^l\|^2 \tag{34}$$

□

The proof of the proposition is given in Appendix E and is based on the technical result of Lemma 16 (Appendix D). More than the upper bound itself (which is fairly rough), Proposition 5 is important to understand the factors influencing the quality of $\hat{\boldsymbol{x}}$. In particular, the bound enables two theoretical predictions about $\hat{\boldsymbol{x}}$. The first prediction is that we expect the quality of the approximation to degrade when increasing the ratio $\frac{\sigma_\delta}{\sigma_\Delta}$ (i.e., when the orientation measurements are more noisy than the position measurements). The second prediction is that $\hat{\boldsymbol{x}}$ will be less accurate for pose graphs having larger inter-nodal distances (recall that $\|\boldsymbol{\Delta}^l\|^2 = \sum_{(i,j)\in\mathcal{E}} \|\boldsymbol{\Delta}_{ij}^l\|^2$, and $\|\boldsymbol{\Delta}_{ij}^l\|$ is the distance between nodes $i$ and $j$).

Fortunately, common SLAM instances are characterized by a small ratio $\frac{\sigma_\delta}{\sigma_\Delta}$ (orientation measurements are usually much more accurate than the position measurements); moreover, the inter-nodal distances $\|\boldsymbol{\Delta}_{ij}\|$, are naturally constrained by sensor range, so they are likely to be small. For these reasons, the suboptimal estimate $\hat{\boldsymbol{x}}$ is expected to be accurate; finally, since the output of LAGO is the estimate $\boldsymbol{x}^\star$, which, according to Proposition 4, is an improved version of $\hat{\boldsymbol{x}}$, the proposed approximation is expected to be close to the optimal solution.

In the experimental section, we will extensively demonstrate that in common problem instances the proposed approximation attains the same objective value of an iterative method in which the local optimization step is repeated several times to correct an inaccurate initial guess (e.g., odometry). Moreover, we show that, when considering more challenging setups (e.g., higher orientation noise), the accuracy of the approximation degrades gracefully.

**Remark 6** (Intrinsic bootstrapping in LAGO)**.** *We already noticed that the first phase in* LAGO *essentially computes an initial guess $\hat{\boldsymbol{\theta}}$ for the orientations. The intuition that a good initial estimate is crucial for fast convergence (and to avoid local minima) has already appeared in related work. A recent example is the work [34], in which Konolige et al. propose to bootstrap node poses by constructing a spanning tree on the graph, using a breadth-first visit. Notice that in [34] the error in the initial guess accumulates when traversing the tree from the root to the leaves. The advantage in* LAGO*, instead, is related to the fact that, by considering all (orientation) measurements at the same time (and not only the ones in the spanning tree), it is able to enforce global correctness of the orientation estimate.* □

### *E. A probabilistic interpretation*

The proposed approximation has a natural probabilistic interpretation. In order to introduce this probabilistic perspective, we recall that the maximum likelihood estimate maximizes the following objective:

$$\boldsymbol{x}^{\mathrm{ML}} = \underset{\boldsymbol{p},\boldsymbol{\theta}}{\arg\max} \;\; \mathbb{P}\left(\boldsymbol{\delta}, \boldsymbol{\Delta}^l \mid \boldsymbol{p}, \boldsymbol{\theta}\right), \tag{35}$$

where $\mathbb{P}\left(\boldsymbol{\delta}, \boldsymbol{\Delta}^l \mid \boldsymbol{p}, \boldsymbol{\theta}\right)$ is the likelihood of measurements $\boldsymbol{\delta}$ and $\boldsymbol{\Delta}^l$ assuming that nodes configuration is $(\boldsymbol{p}, \boldsymbol{\theta})$. Assuming independence between position and orientation measurements we can factorize the likelihood as:

$$\boldsymbol{x}^{\mathrm{ML}} = \underset{\boldsymbol{p},\boldsymbol{\theta}}{\arg\max} \;\; \mathbb{P}\left(\boldsymbol{\Delta}^l \mid \boldsymbol{p}, \boldsymbol{\theta}\right) \;\; \mathbb{P}\left(\boldsymbol{\delta} \mid \boldsymbol{p}, \boldsymbol{\theta}\right). \tag{36}$$

Now we notice that the relative orientation measurements are independent on robot positions, which allows to simplify the second factor as follows:

$$\boldsymbol{x}^{\mathrm{ML}} = \underset{\boldsymbol{p},\boldsymbol{\theta}}{\arg\max} \;\; \mathbb{P}\left(\boldsymbol{\Delta}^l \mid \boldsymbol{p}, \boldsymbol{\theta}\right) \;\; \mathbb{P}\left(\boldsymbol{\delta} \mid \boldsymbol{\theta}\right). \tag{37}$$

Equations (37) contains the basic idea underlying our approximation: instead of optimizing the nonlinear function (37) we first optimize the function $\mathbb{P}\left(\boldsymbol{\delta} \mid \boldsymbol{\theta}\right)$, and then we use such suboptimal solution as an initial guess to optimize the original objective. In fact, it is easy to see that $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathbb{P}\left(\boldsymbol{\delta} \mid \boldsymbol{\theta}\right)$, i.e., $\hat{\boldsymbol{\theta}}$ (the output of Phase 1 in LAGO) maximizes the likelihood of the orientation measurements (thus neglecting the relative position measurements).

If we apply the Bayes rule to the second factor in (37), and we assume flat priors, we obtain $\mathbb{P}\left(\boldsymbol{\delta} \mid \boldsymbol{\theta}\right) \propto \mathbb{P}\left(\boldsymbol{\theta} \mid \boldsymbol{\delta}\right)$, hence:

$$\boldsymbol{x}^{\mathrm{ML}} = \underset{\boldsymbol{p},\boldsymbol{\theta}}{\arg\max} \;\; \mathbb{P}\left(\boldsymbol{\Delta}^l \mid \boldsymbol{p}, \boldsymbol{\theta}\right) \;\; \mathbb{P}\left(\boldsymbol{\theta} \mid \boldsymbol{\delta}\right). \tag{38}$$

Since the measurement model linking $\boldsymbol{\delta}$ to $\boldsymbol{\theta}$ is linear, the posterior $\mathbb{P}\left(\boldsymbol{\theta} \mid \boldsymbol{\delta}\right)$ is a Gaussian distribution, with mean $\hat{\boldsymbol{\theta}}$. Therefore, equation (38) shows how the original maximum likelihood problem over $\boldsymbol{\delta}$ and $\boldsymbol{\Delta}^l$ (with flat prior) can be reduced to a maximum likelihood problem over $\boldsymbol{\Delta}^l$ with a Gaussian prior on the orientations. The interpretation of our approximation easily follows: in the first phase we obtain the Gaussian prior in (38); then in the third phase we approximate the objective (38) around this prior, and we take a Gauss-Newton step towards a maximum of the likelihood function.

## V. Regularization of the Relative Orientation Measurements

We now discuss a crucial point of the proposed approach, that is connected to the periodic nature of the angular information, i.e., robot orientations are defined up to $2k\pi,\ k \in \mathbb{Z}$. Let us introduce the discussion with an example: consider a simple scenario, in which a robot travels along a circumference (in anticlockwise direction) coming back to the starting point. In a noiseless case, summing up all the relative orientation

measurements from the one taken with respect to the first node, to the loop closing constraint, referred to the last node, we obtain $2\pi$. This is because we sum small angular measurements which are defined in $(-\pi, \pi]$. However, the loop closing constraint is expected to link the last pose with the initial pose, whose orientation was set by convention to zero. The linear estimation framework presented so far cannot recognize that the angles $0$ and $2\pi$ actually correspond to the same orientation, hence tries to impose contrasting constraints, producing meaningless results.

This issue stems from the measurement model we assumed in equation (1). In particular, we assumed that the relative orientation measurements follow the linear model:

$$\delta_{ij} = \theta_j - \theta_i, \tag{39}$$

where we omitted measurement noise (we will discuss the influence of noise at the end of this section). However, the model in (39) does not consider that orientation measurements are defined modulo $2\pi$, since angles belong to the manifold $\mathrm{SO}(2)$. Therefore, a more realistic measurement model is

$$\bar{\delta}_{ij} = \langle \theta_j - \theta_i \rangle_{2\pi}\,, \tag{40}$$

where $\langle\cdot\rangle_{2\pi} : \mathbb{R} \to (-\pi, +\pi]$ is the modulus $2\pi$ operator that maps a real number ($\theta_j - \theta_i$ in our case) to the interval $(-\pi, +\pi]$. We can write the modulus operator in explicit form:

$$\bar{\delta}_{ij} = \langle \theta_j - \theta_i \rangle_{2\pi} = \theta_j - \theta_i + 2k_{ij}\pi, \tag{41}$$

where $k_{ij}$ is a suitable integer that forces $\bar{\delta}_{ij}$ in the interval $(-\pi, +\pi]$. We shall call $k_{ij}$ a *regularization term*.

If we are able to compute the regularization term $k_{ij}$ we can easily convert (41) into (39): once we know $k_{ij}$, we can compute the *regularized measurements*

$$\delta_{ij} \doteq \bar{\delta}_{ij} - 2k_{ij}\pi = \theta_j - \theta_i, \tag{42}$$

coming back to the linear model (1). We call $\delta_{ij}$ *regularized orientation measurements*, to distinguish them from the measured relative orientations $\bar{\delta}_{ij}$.

In this section we discuss how to compute $k_{ij},\ (i,j) \in \mathcal{E}$. The key insight is that in a linear model in the form $\delta_{ij} = \theta_j - \theta_i,\ (i,j) \in \mathcal{E}$, the relative measurements $\delta_{ij}$ have to sum-up to zero along each cycle in the graph; this property will be lately referred to as *zero-sum* property. Therefore, we look for the $k_{ij},\ (i,j) \in \mathcal{E}$ in (42) that allow to satisfy the zero-sum property. Before presenting the main result (Theorem 11) let us introduce some specific concepts from graph theory [32].

A *cycle* is a subgraph in which every node appears in an even number of edges. A *circuit* is a cycle in which every node appears exactly in two edges. We can represent a (*directed*) circuit as a vector $\boldsymbol{c}_i$ of $m$ elements in which the $k$-th element is $+1$ or $-1$ if edge $k$ belongs to the circuit and it is traversed respectively forwards (from tail to head) or backwards, and $0$ if it does not appear in the circuit (notice that the ordering of the edges in $\boldsymbol{c}_i$ is arbitrary). A pictorial example is reported in Figure 2.

**Definition 7** (Fundamental circuit)**.** *Given a directed graph $\mathcal{G}$ and a spanning tree $\mathsf{T}$ of $\mathcal{G}$, a* fundamental circuit *is a circuit composed by a chord $(i,j)$ of $\mathcal{G}$ with respect to $\mathsf{T}$ and the unique path in $\mathsf{T}$ connecting $i$ and $j$.* □

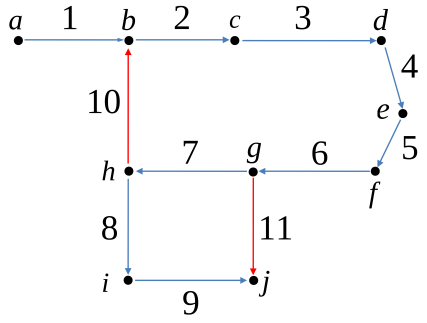



Fig. 2. Pose graph with odometric constraints (in blue) and loop closing constraints (in red). We number the edges from 1 to 11. A circuit in the graph includes edges $\{2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 10\}$ and is described by the vector $\boldsymbol{c}_1 = [0\ +1\ +1\ +1\ +1\ +1\ 0\ 0\ +1\ 0]^\mathsf{T}$. $\boldsymbol{c}_1$ is also a fundamental circuit (Definition 7), since, considering the (odometric) spanning tree $\mathsf{T} = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9\}$, $\boldsymbol{c}_1$ includes a chord (edge 10) and edges belonging to $\mathsf{T}$. A second (fundamental) circuit is $\{7,\ 8,\ 9,\ 11\}$ and is described by the vector $\boldsymbol{c}_2 = [0\ 0\ 0\ 0\ 0\ 0\ +1\ +1\ +1\ 0\ -1]^\mathsf{T}$ (edge 11 is traversed backwards). A cycle basis matrix $\boldsymbol{C}$ of the graph has a number of rows equal to the number of loop closing constraints. According to Theorem 8 we can build a cycle basis matrix from the fundamental circuits, i.e., $\boldsymbol{C} = [\boldsymbol{c}_1\ \boldsymbol{c}_2]^\mathsf{T}$.

A *cycle basis* of $\mathcal{G}$ is the smallest set of circuits such that any cycle in the graph can be written as a combination of the circuits in the basis. The space spanned by the vectors in the basis is called *cycle space*.

**Theorem 8** (Fundamental cycle basis)**.** *The set of the fundamental circuits of a graph $\mathcal{G}$ forms a cycle basis of $\mathcal{G}$.* □

The proof of the previous theorem can be found in several books of graph theory, see, e.g., [13]. We already mentioned that a spanning tree $\mathsf{T}$ of a connected graph $\mathcal{G}$ contains exactly $n$ edges. Accordingly, the number of chords, hence of fundamental circuits in $\mathcal{G}$, is $m - n$.

**Corollary 9** (Dimension of the cycle space)**.** *The dimension of the cycle space of a connected graph $\mathcal{G}$ is $\ell = m - n$, and it is called* cyclomatic *number of the graph [32].* □

**Corollary 10** (Structure of the fundamental cycle basis matrix)**.** *Ordering the edges of a connected directed graph $\mathcal{G}$ from $1$ to $m$, so that the first $n$ elements are the edges of a given spanning tree $\mathsf{T}$, and the last $m - n$ elements are the chords of $\mathcal{G}$ with respect $\mathsf{T}$, the matrix containing all the vectors $\boldsymbol{c}_i$ corresponding to the fundamental circuits can be written as:*

$$\boldsymbol{C} = [\boldsymbol{c}_1\ \boldsymbol{c}_2\ \ldots\ \boldsymbol{c}_\ell]^\mathsf{T} = [\boldsymbol{D}\quad \mathbf{I}_\ell]\,, \tag{43}$$

*where $\mathbf{I}_\ell$ is the identity matrix of dimension $\ell$ and $\boldsymbol{D}$ is a matrix with elements in $\{-1, 0, 1\}$. $\boldsymbol{C}$ is usually referred to as* (fundamental) cycle basis matrix. □

The previous result is a direct consequence of the structure of the fundamental circuits, each one containing a single chord and a collection of edges in the spanning tree [13]. With slight abuse of notation, in the following, $\boldsymbol{C}$ will denote both the

cycle basis and the cycle basis matrix. Figure 2 shows an example of fundamental cycle basis matrix.

According to the machinery introduced so far, we notice that the zero-sum property essentially requires that the sum of the relative orientation measurements along every cycle in the graph is zero, instead of $2k\pi, k \in \mathbb{Z}\backslash\{0\}$. Hence we can state the following theorem that holds under the assumption of noiseless angular measurements.

**Theorem 11** (Existence of regularization terms)**.** *Given relative orientation measurements $\bar{\boldsymbol{\delta}} = [\bar{\delta}_1 \ \bar{\delta}_2 \ \dots \ \bar{\delta}_m]^\mathsf{T}$, such that the first $n$ elements correspond to the edges of a given spanning tree $\mathsf{T}$, and the last $m-n$ elements correspond to the chords of $\mathcal{G}$ with respect $\mathsf{T}$, there exists a regularization vector $\boldsymbol{k} = [k_1 \ k_2 \ \dots \ k_m]^\mathsf{T}$ so that the regularized orientation measurements $\boldsymbol{\delta} = [\bar{\delta}_1 - 2\pi k_1 \ \bar{\delta}_2 - 2\pi k_2 \ \dots \ \bar{\delta}_m - 2\pi k_m]^\mathsf{T}$ satisfy the zero-sum property.*

**Proof.** Let us start by formulating the zero-sum property in a more familiar way. A necessary and sufficient condition for the zero-sum property to be satisfied for all the cycles in the graph is that it is satisfied for the cycles in the cycle basis [32]. Let us consider the cycle basis composed by the fundamental circuits; for the zero-sum property to hold true, the corrected measurements $\boldsymbol{\delta}$ have to satisfy:

$$\boldsymbol{c}_i^\mathsf{T}\boldsymbol{\delta} = 0 \quad \forall \boldsymbol{c}_i \in \boldsymbol{C}. \tag{44}$$

Roughly speaking, if the sum of the relative orientation measurements is zero for the edges in the fundamental circuits, this property is true for every cycle in the graph. Equation (44) can be written in compact form using the cycle basis matrix:

$$\boldsymbol{C}\boldsymbol{\delta} = \mathbf{0}_\ell. \tag{45}$$

According to the definition of regularized measurements we can rewrite (45) as:

$$\boldsymbol{C}(\bar{\boldsymbol{\delta}} - 2\pi\boldsymbol{k}) = \mathbf{0}_\ell \Longrightarrow \boldsymbol{C}\boldsymbol{k} = \frac{1}{2\pi}\boldsymbol{C}\bar{\boldsymbol{\delta}}. \tag{46}$$

The right hand side will contain the sum of the original measurements for each fundamental circuit. In a noiseless case, the vector $\boldsymbol{C}\bar{\boldsymbol{\delta}}$ contains terms in the form $2k\pi, \ k \in \mathbb{Z}$, therefore $\frac{1}{2\pi}\boldsymbol{C}\bar{\boldsymbol{\delta}}$ will be an integer vector. Since the cycle basis matrix $\boldsymbol{C}$ can be computed from the graph and $\bar{\boldsymbol{\delta}}$ is a given of the problem, the only unknown of (46) is $\boldsymbol{k}$ and the existence of suitable regularization terms is reduced the demonstration of existence of a solution to system (46). Recalling (43), it is easy to show that a solution to system (46), is $\boldsymbol{k} = \left[\ \mathbf{0}_n^\mathsf{T} \ (\frac{1}{2\pi}\boldsymbol{C}\bar{\boldsymbol{\delta}})^\mathsf{T}\ \right]^\mathsf{T}$:

$$\boldsymbol{C}\boldsymbol{k} = [\boldsymbol{D} \ \ \mathbf{I}_\ell]\left[\mathbf{0}_n^\mathsf{T} \ (\frac{1}{2\pi}\boldsymbol{C}\bar{\boldsymbol{\delta}})^\mathsf{T}\right]^\mathsf{T} = \frac{1}{2\pi}\boldsymbol{C}\bar{\boldsymbol{\delta}}, \tag{47}$$

hence proving our thesis. □

The process of compensating the relative orientation measurements is named *regularization*. The proof of Theorem 11 also provides a viable approach to compute the regularization terms: choosing $\boldsymbol{k} = \left[\ \mathbf{0}_n^\mathsf{T} \ (\frac{1}{2\pi}\boldsymbol{C}\bar{\boldsymbol{\delta}})^\mathsf{T}\ \right]^\mathsf{T}$ assures that the regularized orientation measurements satisfy the zero-sum property. We notice that the aforementioned solution only requires to correct the angular measurements corresponding to the chords, without any modification to the edges in the spanning tree (the first $n$ entries in $\boldsymbol{k}$ are zeros). Recalling Lemma 1, if we consider the spanning tree comprising the odometric constraints, we can regularize the relative orientation measurements by simply correcting loop closing constraints; in particular, a loop closing measurement, constraining two robot poses, has to be corrected taking into account the number of complete turns ($2\pi$ turns) the robot did when traveling from the first to the second pose, with the sign of the regularization terms being consistent with the convention chosen for the orientation measurements.

For the derivation of this section we assumed noiseless measurements. In the case of noisy relative measurements, the zero sum property (45) cannot be met exactly: LAGO, in fact, will be in charge of compensating the measurement errors by minimizing a suitable cost function. Accordingly, the term $\boldsymbol{C}\bar{\boldsymbol{\delta}}$ in (46) will not contain exact multiples of $2\pi$, and $\frac{1}{2\pi}\boldsymbol{C}\bar{\boldsymbol{\delta}}$ will not be an integer vector. However, we found that in practice a simple rounding of $\frac{1}{2\pi}\boldsymbol{C}\bar{\boldsymbol{\delta}}$ allows to retrieve the desired regularization terms. One may argue that, if the noise is large, it is not possible to discern the desired regularization terms, since the rounding cannot compensate measurement errors; however this issue was not found to be relevant in common applications (see experimental section): the impossibility to determine the correct multiple of $2\pi$ means that the amount of noise is so high that the robot, revisiting a past pose, is not able to discern how many times the robot turned around itself (i.e., completed $2\pi$ turns) since the previous visit. Note that this result also sheds some light on the so called *orientation wraparound* problem [24], which is known to prevent convergence in iterative approaches.

## VI. Experiments and Discussion

In this section we present an experimental evaluation of LAGO. Section VI-A describes a practical implementation of the approach. Section VI-B presents the techniques compared in the experiments and the performance metrics. Section VI-C shows results of LAGO and related techniques in standard benchmarking datasets. Section VI-D evaluates the approaches on more challenging simulated data. Both simulations and real tests highlight that LAGO assures a remarkable computational advantage, while being very accurate in practice.

### A. Implementation

In a real problem, the proposed SLAM approach involves the following steps:

1) *Information synthesis* (SLAM front-end): odometric and loop closing constraints are extracted from sensor data;
2) *Regularization*: regularization terms are added to loop closing constraints so that the zero-sum property is satisfied;
3) *Linear Approximation*: LAGO is applied to solve the graph embedding problem using the regularized orientation measurements.

As mentioned before, the front-end is sensor-dependent and is not treated in this paper (we consider the constraints in the pose graph as *input data* for our problem). The *regularization*

is described in Section V and reported as a pseudo-code in Algorithm 1.

**Algorithm 1:** Regularization

input:
 reduced incidence matrix $\boldsymbol{A}$ (topology of the graph)
 (original) orientation measurements $\bar{\boldsymbol{\delta}}$
output:
 regularized orientation measurements $\boldsymbol{\delta}$

# Compute a fundamental cycle basis matrix
$\boldsymbol{C} \leftarrow$ COMPUTE-CYCLE-BASIS$(\boldsymbol{A})$
# Compute regularization terms
$\hat{\boldsymbol{k}} = \left[\mathbf{0}_n^\mathsf{T} \ (\frac{1}{2\pi}\boldsymbol{C}\bar{\boldsymbol{\delta}})^\mathsf{T}\right]^\mathsf{T}$;
$\boldsymbol{k} = \texttt{round}(\hat{\boldsymbol{k}})$;
# Regularize orientation measurements
$\boldsymbol{\delta} = \bar{\boldsymbol{\delta}} - 2\pi\boldsymbol{k}$.
**return** $\boldsymbol{\delta}$.

External functions:
- COMPUTE-CYCLE-BASIS$(\boldsymbol{A})$ computes a fundamental cycle basis for the graph $\mathcal{G}$, according to Corollary 10.

After performing regularization, it is possible to apply LAGO. From Section IV, one may draw the following conclusions: LAGO requires to compute $\boldsymbol{P}_{\hat{\boldsymbol{\theta}}} = (\boldsymbol{A}\boldsymbol{P}_\delta^{-1}\boldsymbol{A}^\mathsf{T})^{-1}$, to be included in the covariance propagation of Phase 2, and $\boldsymbol{P}_{\boldsymbol{z}}^{-1}$, which is used in equation (24). In this section we show that a suitable implementation of the approach requires no matrix inversion at all. The pseudocode of LAGO is summarized in Algorithm 2. The key insight is that some steps of LAGO can be performed in *information form*, instead of using the corresponding covariance matrices. For instance, the computation of $\boldsymbol{x}^\star$ in (24) requires the knowledge of $\boldsymbol{P}_{\boldsymbol{z}}^{-1}$ rather than $\boldsymbol{P}_{\boldsymbol{z}}$; therefore we can directly use the matrix $\boldsymbol{\Omega}_{\boldsymbol{z}} \doteq \boldsymbol{P}_{\boldsymbol{z}}^{-1}$, given in (25). This choice also implies a second advantage: since $\boldsymbol{P}_{\boldsymbol{z}}^{-1}$ in (25) only contains the information matrix $\boldsymbol{P}_{\hat{\boldsymbol{\theta}}}^{-1}$, we do not need to compute the matrix $\boldsymbol{P}_{\hat{\boldsymbol{\theta}}}$, but we can work directly on the information matrix $\boldsymbol{\Omega}_{\hat{\boldsymbol{\theta}}} \doteq \boldsymbol{P}_{\hat{\boldsymbol{\theta}}}^{-1} = \boldsymbol{A}\boldsymbol{\Omega}_{\boldsymbol{\delta}}\boldsymbol{A}^\mathsf{T}$, thus avoiding matrix inversion.

In Algorithm 2, the function `linear_solver`$(\boldsymbol{A}, \boldsymbol{b})$ contains a standard linear equation solver of system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, while the content of the routines for building the involved matrices (`build_R`, `build_J`) can be easily derived from Section IV.

**Remark 12** (Complexity)**.** *A generic Gauss-Newton method requires solving $N$ linear systems of size $3n$ (in which the number of iterations $N$ may increase arbitrarily, depending on the quality of the initial guess); the cost of the proposed approach is due to (i)* regularization*, which requires to fill in the matrix $\boldsymbol{C}$ and to perform a (sparse) vector-matrix multiplication, (ii)* linear approximation*, which requires to solve a smaller problem (on robot orientations - size $n$) and a problem of the same size of a single step of the Gauss-Newton method. Therefore the asymptotic complexity of* LAGO *reduces to the solution of a sparse linear system of size $3n$.* □

An example of the sparsity patterns of the matrices involved in the linear systems of line 12 and 25 of Algorithm 2 is reported in Figure 3.

**Algorithm 2:** LAGO

input:
 reduced incidence matrix $\boldsymbol{A}$ (topology of the graph)
 regularized orientation measurements $\boldsymbol{\delta}$
 information matrix of the orientation measurements $\boldsymbol{\Omega}_{\boldsymbol{\delta}}$
 relative position measurements $\boldsymbol{\Delta}^l$
 information matrix of position measurements $\boldsymbol{\Omega}_{\boldsymbol{\Delta}^l}$
output:
 estimate of graph configuration $\boldsymbol{x}^\star$

# Phase 1
$\boldsymbol{\Omega}_{\hat{\boldsymbol{\theta}}} = \boldsymbol{A}\boldsymbol{\Omega}_{\boldsymbol{\delta}}\boldsymbol{A}^\mathsf{T}$
$\hat{\boldsymbol{\theta}} = \texttt{linear\_solver}(\boldsymbol{\Omega}_{\hat{\boldsymbol{\theta}}}\ ,\ \boldsymbol{A}\boldsymbol{\Omega}_{\boldsymbol{\delta}}\boldsymbol{\delta})$

# Phase 2
$\hat{\boldsymbol{R}} = \texttt{build\_R}(\hat{\boldsymbol{\theta}})$
$\boldsymbol{\Omega}_{\boldsymbol{\Delta}^g} = \hat{\boldsymbol{R}}\boldsymbol{\Omega}_{\boldsymbol{\Delta}^l}\hat{\boldsymbol{R}}^\mathsf{T}$
$\boldsymbol{J} = \texttt{build\_J}(\hat{\boldsymbol{\theta}}, \boldsymbol{\Delta}^l)$
$\boldsymbol{\Omega}_{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{\Omega}_{\boldsymbol{\Delta}^g} & -\boldsymbol{\Omega}_{\boldsymbol{\Delta}^g}\boldsymbol{J} \\ -\boldsymbol{J}^\mathsf{T}\boldsymbol{\Omega}_{\boldsymbol{\Delta}^g} & \boldsymbol{\Omega}_{\hat{\boldsymbol{\theta}}} + \boldsymbol{J}^\mathsf{T}\boldsymbol{\Omega}_{\boldsymbol{\Delta}^g}\boldsymbol{J} \end{bmatrix}$
$\boldsymbol{z} = [(\hat{\boldsymbol{R}}\boldsymbol{\Delta}^l)^\mathsf{T}\ \hat{\boldsymbol{\theta}}^\mathsf{T}]^\mathsf{T}$

# Phase 3
$\boldsymbol{A}_2 = \boldsymbol{A} \otimes \mathbf{I}_2$
$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A}_2 & \mathbf{0}_{2n\times n} \\ \mathbf{0}_{n\times 2m} & \mathbf{I}_n \end{bmatrix}$
$\boldsymbol{\Omega}_{\boldsymbol{x}^\star} = \boldsymbol{B}\boldsymbol{\Omega}_{\boldsymbol{z}}\boldsymbol{B}^\mathsf{T}$
$\boldsymbol{x}^\star = \texttt{linear\_solver}(\boldsymbol{\Omega}_{\boldsymbol{x}^\star}\ ,\ \boldsymbol{B}\boldsymbol{\Omega}_{\boldsymbol{z}}\boldsymbol{z})$

**return** $\boldsymbol{x}^\star$.

External functions:
- `linear_solver`$(\boldsymbol{A}, \boldsymbol{b})$ solves a sparse linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$;
- `build_R`$(\hat{\boldsymbol{\theta}})$ builds the matrix $\hat{\boldsymbol{R}}$;
- `build_J`$(\hat{\boldsymbol{\theta}}, \boldsymbol{\Delta}^l)$ builds the Jacobian appearing in (17).

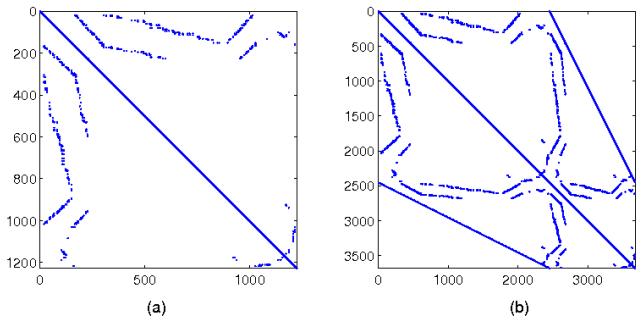



Fig. 3. Sparsity patterns of the matrices involved in Phase 1 and 3 of the proposed approach for the INTEL scenario (see Section VI-B): (a) $\boldsymbol{\Omega}_{\hat{\boldsymbol{\theta}}}$ (line 11, Algorithm 2); (b) $\boldsymbol{\Omega}_{\boldsymbol{x}^\star}$ (line 24, Algorithm 2).

### B. Compared techniques and performance metrics

In the following sections we benchmark LAGO towards state-of-the-art approaches for pose graph optimization. We compare the following algorithms:

LAGO is the approach proposed in this paper, implemented as in Section VI-A. A c++ implementation of LAGO (including regularization) is available online at [7].

Toro **[24]** is based on gradient-descent and uses a tree-based parametrization to distribute (in a possibly suboptimal

way) residual errors across the graph. We use a standard implementation of Toro (available at [50]) with default settings (100 iterations).

g2o **[35]** uses an on-manifold implementation of a Gauss-Newton method. We use a standard implementation of g2o (available at [50]) with default settings (5 iterations).

g2o10 is the same as g2o but we set the code to perform 10 Gauss-Newton iterations.

iSAM **[29]–[31]** uses a QR factorization within an iterative optimization scheme. We report the comparison with iSAM for completeness, although iSAM is actually optimized to solve incremental SLAM problems. We use a standard implementation of iSAM (available at [50]) which also allows to perform batch optimization.

The accuracy of the approaches is evaluated by looking at the $\chi^2$ value, which corresponds to the value of the objective function (2) to be minimized (therefore, smaller $\chi^2$ means higher accuracy). The computational effort is evaluated by measuring the CPU time the approaches take to solve pose graph optimization. We tested the approaches on a desktop computer with processor *INTEL Core 2 Quad*, 2.83 GHz. The $\chi^2$ values and the CPU times are averaged over 100 runs.

### C. Experiments on standard benchmarking datasets

We tested the performance of the techniques described in the previous section on the following benchmarking datasets:

FR079**:** This dataset was acquired at the *Freiburg Building* 079. Constraints in the pose graph are obtained from [37]. The pose graph has 989 nodes and 1217 edges.

CSAIL**:** This dataset was acquired at the MIT *CSAIL Building*. The corresponding pose graph has 1045 nodes and 1172 edges [37].

FRH**:** This dataset was acquired at the *Freiburg University Hospital*. The corresponding pose graph has 1316 nodes and 2820 edges [37].

INTEL**:** This dataset, acquired at the *Intel Research Lab* in Seattle[2], includes odometry and laser range-finder data. Odometric constraints are derived from a scan matching algorithm, see [36], while loop closing constraints are selected from the relations available at [37]. The corresponding pose graph has 1228 nodes and 1505 edges.

M3500**:** This simulated dataset was created by Olson *et al.* [43]. The corresponding pose graph has 3500 nodes and 5453 edges.

M10000**:** This simulated dataset is an extended version of the M3500 dataset [23]. The corresponding pose graph has 10000 nodes and 64311 edges.

We tested Toro, g2o, iSAM, and LAGO on the same set of constraints and with the same measurement covariance matrices. For the iterative techniques (excluding iSAM, which autonomously builds an initial estimate), the odometric trajectory is used as initial guess. The results of the tests are reported in Table I, while the corresponding pose graphs, estimated with LAGO, are shown in Figures 4-5.

[2]The dataset is provided by Dirk Hähnel and available online [37].

Let us discuss the results of Table I. We considered two different hypotheses for measurement covariance: the identity matrix (rows denoted with $\mathbf{I}$ in Table I) and a block diagonal covariance matrix (rows denoted with $\boldsymbol{P}$ in Table I) in which larger displacements correspond to higher uncertainty. All the input files used in the tests can be found online [7]. In all scenarios (except M10000) g2o and g2o10 attain the same objective: the benchmarking scenarios have a quite accurate initial guess and in most cases few Gauss-Newton iterations enable convergence to a minimum of the cost function. Only in the scenario M10000 performing more iterations in g2o improves the accuracy. As expected, g2o10 takes about twice the time of g2o, as it performs 10 iterations instead of 5.

Also iSAM is able to converge in few iterations (last column of Table I). It is worth noticing that the stopping condition in iSAM is not imposed on the number of iterations (as in Toro and g2o), but the approach terminates when the local correction is sufficiently small. The accuracy of iSAM is comparable with the one of g2o, although in the scenarios with non-diagonal covariance matrices (rows labelled with $\boldsymbol{P}$) iSAM performs slightly worse. In the tested scenarios iSAM is $3-10$ times slower than g2o.

Toro is inferior to all other techniques in terms of accuracy (due to the involved approximation) and is remarkably slower. In the following section, however, we will see that in more challenging scenarios it has some advantages, as it is less prone to converge to local minima.

The accuracy of LAGO is very close to g2o, g2o10, and iSAM. In the scenarios FR079, CSAIL, FRH, and M3500 there is practically no difference between LAGO and g2o, while LAGO tends to be slightly more accurate than iSAM. In the scenario INTEL-$\boldsymbol{P}$ the accuracy of LAGO is inferior. We impute this issue to the fact that the choice of the covariance matrix can "amplify" residual errors and also a small error in few edges can lead to a large $\chi^2$ value. A second reason is that the orientation measurements of this dataset are less accurate and this may degrade the performance of LAGO as we will remark in the following section. Notice that the estimated pose graph is still visually correct, see Figure 4(a). Finally, we notice that in the large-scale scenario M10000, LAGO is more accurate than g2o and iSAM and attains the same objective of g2o10. The CPU times show that LAGO is one order of magnitude faster than iSAM and two times faster than g2o. In all the test, the suboptimality gap provided in Proposition 5 is a gross upper bound as it is at least two orders of magnitude larger than the actual gap (assuming that g2o10 attains the optimal solution).

It is now clear that the proposed approach can perform at least as well as the most accurate state-of-the-art approaches to SLAM, although it requires a fraction of their computational effort. In the next section we will consider more challenging problem instances, in order to point out the limitations of the proposed approximation.

We conclude this experimental section by observing that loop closing constraints in a pose graph are not necessarily connected to place revisiting episodes. An interesting alternative is the case in which loop closing constraints correspond to GPS information: in this case the measurements are referred to an absolute reference frame (set as the first node in the

| | | | LAGO | Toro | g2o | g2o10 | iSAM | |
|---|---|---|---|---|---|---|---|---|
| FR079 | $\mathbf{I}$ | $\chi^2$ | $7.20 \cdot 10^{-2}$ | $8.60 \cdot 10^{-2}$ | $7.19 \cdot 10^{-2}$ | $7.19 \cdot 10^{-2}$ | $7.19 \cdot 10^{-2}$ | 3 iter. |
| | | Time | $5.93 \cdot 10^{-3}$ | $2.95 \cdot 10^{-1}$ | $9.69 \cdot 10^{-3}$ | $1.74 \cdot 10^{-2}$ | $4.98 \cdot 10^{-2}$ | |
| | $\boldsymbol{P}$ | $\chi^2$ | $3.94 \cdot 10^{+1}$ | $4.74 \cdot 10^{+2}$ | $3.87 \cdot 10^{+1}$ | $3.87 \cdot 10^{+1}$ | $4.88 \cdot 10^{+1}$ | 3 iter. |
| | | Time | $5.81 \cdot 10^{-3}$ | $2.95 \cdot 10^{-1}$ | $9.65 \cdot 10^{-3}$ | $1.73 \cdot 10^{-2}$ | $5.01 \cdot 10^{-2}$ | |
| CSAIL | $\mathbf{I}$ | $\chi^2$ | $1.07 \cdot 10^{-1}$ | $1.17 \cdot 10^{-1}$ | $1.07 \cdot 10^{-1}$ | $1.07 \cdot 10^{-1}$ | $1.07 \cdot 10^{-1}$ | 3 iter. |
| | | Time | $5.67 \cdot 10^{-3}$ | $2.80 \cdot 10^{-1}$ | $9.28 \cdot 10^{-3}$ | $1.67 \cdot 10^{-2}$ | $4.84 \cdot 10^{-2}$ | |
| | $\boldsymbol{P}$ | $\chi^2$ | $4.06 \cdot 10^{+1}$ | $2.41 \cdot 10^{+3}$ | $4.06 \cdot 10^{+1}$ | $4.06 \cdot 10^{+1}$ | $1.17 \cdot 10^{+2}$ | 3 iter. |
| | | Time | $5.75 \cdot 10^{-3}$ | $2.80 \cdot 10^{-1}$ | $9.30 \cdot 10^{-3}$ | $1.67 \cdot 10^{-2}$ | $4.90 \cdot 10^{-2}$ | |
| FRH | $\mathbf{I}$ | $\chi^2$ | $1.39 \cdot 10^{-6}$ | $3.42 \cdot 10^{-4}$ | $1.39 \cdot 10^{-6}$ | $1.39 \cdot 10^{-6}$ | $1.39 \cdot 10^{-6}$ | 1 iter. |
| | | Time | $1.10 \cdot 10^{-2}$ | $3.71 \cdot 10^{0}$ | $2.09 \cdot 10^{-2}$ | $3.89 \cdot 10^{-2}$ | $5.96 \cdot 10^{-2}$ | |
| | $\boldsymbol{P}$ | $\chi^2$ | $1.91 \cdot 10^{-4}$ | $5.23 \cdot 10^{-2}$ | $1.91 \cdot 10^{-4}$ | $1.91 \cdot 10^{-4}$ | $1.91 \cdot 10^{-4}$ | 1 iter. |
| | | Time | $1.10 \cdot 10^{-2}$ | $3.71 \cdot 10^{0}$ | $2.08 \cdot 10^{-2}$ | $3.88 \cdot 10^{-2}$ | $6.07 \cdot 10^{-2}$ | |
| INTEL | $\mathbf{I}$ | $\chi^2$ | $8.07 \cdot 10^{-1}$ | $1.17 \cdot 10^{0}$ | $7.89 \cdot 10^{-1}$ | $7.89 \cdot 10^{-1}$ | $7.89 \cdot 10^{-1}$ | 6 iter. |
| | | Time | $7.00 \cdot 10^{-3}$ | $3.69 \cdot 10^{-1}$ | $1.20 \cdot 10^{-2}$ | $2.16 \cdot 10^{-2}$ | $1.03 \cdot 10^{-1}$ | |
| | $\boldsymbol{P}$ | $\chi^2$ | $1.45 \cdot 10^{+4}$ | $1.03 \cdot 10^{+5}$ | $2.15 \cdot 10^{+2}$ | $2.15 \cdot 10^{+2}$ | $2.24 \cdot 10^{+2}$ | 5 iter. |
| | | Time | $6.91 \cdot 10^{-3}$ | $3.70 \cdot 10^{-1}$ | $1.20 \cdot 10^{-2}$ | $2.16 \cdot 10^{-2}$ | $8.99 \cdot 10^{-2}$ | |
| M3500 | $\mathbf{I}$ | $\chi^2$ | $3.02 \cdot 10^{0}$ | $5.42 \cdot 10^{0}$ | $3.02 \cdot 10^{0}$ | $3.02 \cdot 10^{0}$ | $3.02 \cdot 10^{0}$ | 5 iter. |
| | | Time | $3.18 \cdot 10^{-2}$ | $1.52 \cdot 10^{0}$ | $6.74 \cdot 10^{-2}$ | $1.26 \cdot 10^{-1}$ | $3.78 \cdot 10^{-1}$ | |
| | $\boldsymbol{P}$ | $\chi^2$ | $3.73 \cdot 10^{+3}$ | $2.18 \cdot 10^{+6}$ | $3.55 \cdot 10^{+3}$ | $3.55 \cdot 10^{+3}$ | $4.07 \cdot 10^{+3}$ | 5 iter. |
| | | Time | $3.22 \cdot 10^{-2}$ | $1.55 \cdot 10^{0}$ | $6.84 \cdot 10^{-2}$ | $1.28 \cdot 10^{-1}$ | $3.88 \cdot 10^{-1}$ | |
| M10000 | $\mathbf{I}$ | $\chi^2$ | $3.03 \cdot 10^{+2}$ | $3.29 \cdot 10^{+2}$ | $3.03 \cdot 10^{+2}$ | $3.03 \cdot 10^{+2}$ | $3.03 \cdot 10^{+2}$ | 6 iter. |
| | | Time | $3.52 \cdot 10^{-1}$ | $1.70 \cdot 10^{+1}$ | $6.59 \cdot 10^{-1}$ | $1.26 \cdot 10^{0}$ | $7.81 \cdot 10^{0}$ | |
| | $\boldsymbol{P}$ | $\chi^2$ | $1.99 \cdot 10^{+5}$ | $7.70 \cdot 10^{+6}$ | $2.28 \cdot 10^{+5}$ | $1.99 \cdot 10^{+5}$ | $2.72 \cdot 10^{+5}$ | 7 iter. |
| | | Time | $3.54 \cdot 10^{-1}$ | $1.71 \cdot 10^{+1}$ | $6.60 \cdot 10^{-1}$ | $1.26 \cdot 10^{0}$ | $9.14 \cdot 10^{0}$ | |

TABLE I

OBJECTIVE FUNCTION VALUE ($\chi^2$) AND AVERAGE COMPUTATION TIME (IN SECONDS) FOR THE COMPARED APPROACHES.

pose graph) and the corresponding edges connect the node that takes the GPS measurement with such reference frame. With straightforward generalizations, we can adapt the proposed approach to deal with this absolute information. We tested the proposed approximation on a real dataset from the *Rawseeds* project [3]: odometric constraints are based on wheel odometry, whereas loop closing constraints correspond to accurate measurements from an *RTK*-GPS (for sake of simplicity two consecutive GPS measurements are supposed to identify also robot orientation, due to non-holonomic constraints in robot motion). The odometric configuration and the loop closing constraints are reported in Figure 6(a). The configuration, estimated with the proposed approach, is shown in Figure 6(b) superimposed to the satellite map of the outdoor scenario.

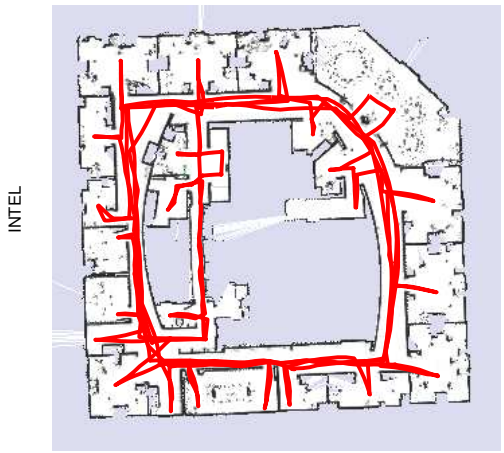
**Remark 13** (Independence assumption)**.** *The assumption of independent position and orientation measurements holds true when dealing with holonomic platforms. For non-holonomic platforms it constitutes an approximation, but several state-of-the-art techniques have been demonstrated to produce effective results, even under stricter assumptions on the covariance structure (e.g. spherical covariances in [24]). In the recent work [12] it has been proposed an approach that relaxes the independence assumption. However, despite the technical complications of dealing with full covariances, the approach in [12] does not entail a remarkable practical advantage with respect to* LAGO. □

### *D. Simulations on large-scale datasets*

Simulations allow to assess accuracy, computational effort, and scalability of the proposed approach in both common and more challenging problem instances (e.g., very large-scale scenarios, unusual measurement noise, etc). We build a simulation scenario as shown in Figure 7. We first construct robot trajectory, considering the case in which the robot, starting from $(0, 0)$, covers a square environment, following a square-waved path. We initialize a node in the pose graph after every travelled meter (nodes are shown as black dots in Figure 7). For each node, with probability $\rho_{\text{lc}}$, we add a loop closing constraint with one of the closest nodes in the pose graph (excluding the nodes that are already connected by odometric constraints). From the real pose graph we build the relative position measurements, adding Gaussian noise with covariance $\boldsymbol{P}_{\boldsymbol{\Delta}^l} = \sigma_{\Delta}^2 \mathbf{I}_{2m}$, and the relative orientation measurements, adding Gaussian noise with covariance $\boldsymbol{P}_{\boldsymbol{\delta}} = \sigma_{\delta}^2 \mathbf{I}_m$.

In the following paragraphs we evaluate the accuracy and the computational effort of LAGO, Toro, g2o, g2o10, and iSAM, in function of measurement uncertainty, number of nodes, and inter-nodal distances.

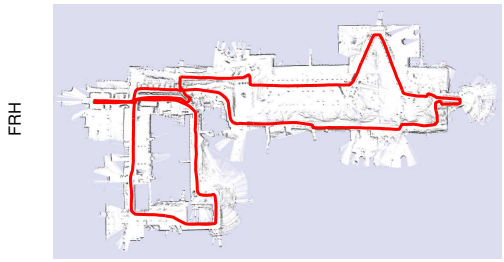
*Influence of orientation measurement uncertainty*: We tested the pose graph optimization approaches for different values of $\sigma_{\delta}$. We considered a pose graph with $10^4$ nodes, and, in each test, we randomly added loop closings with probability $\rho_{\text{lc}} = 0.5$. We tested the following values of orientation un-




(a)


(b)

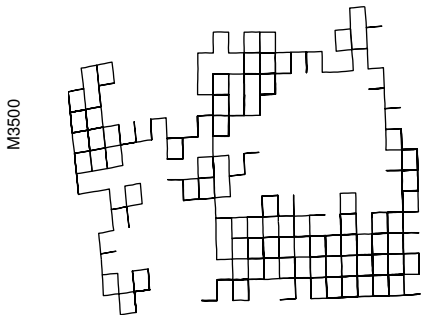

(c)



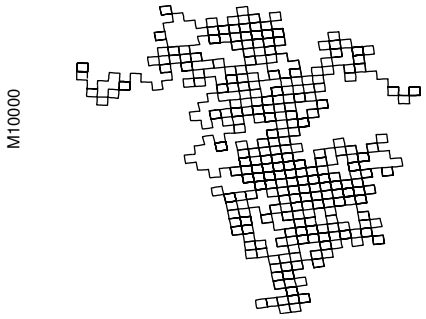

(d)

Fig. 4. Estimated configurations from LAGO superimposed on the reference map available online: (a) Intel Research Lab (INTEL), (b) Freiburg Building 079 (FR079), (c) MIT CSAIL Building (CSAIL), (d) Freiburg University Hospital (FRH).

certainty $\sigma_\delta = \{0.01, 0.03, 0.05, 0.07, 0.1, 0.2\}\text{rad}$, while we kept fixed the position measurement uncertainty $\sigma_\Delta = 0.5\text{m}$. For each value we tested the approaches on 100 realizations of the pose graph. The $\chi^2$ values are shown in Figure 8(a). For small orientation uncertainty ($\leq 0.03\text{rad}$) we find the same

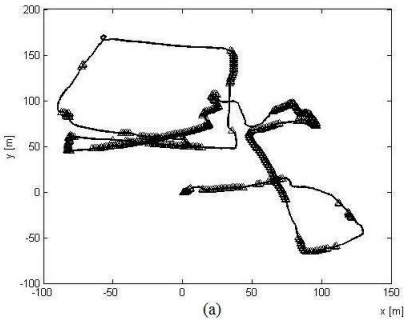



(a)




(b)

Fig. 5. Estimated configurations from LAGO on simulated benchmarking datasets: (a) Manhattan World with 3500 nodes (M3500), (b) Manhattan World with 10000 nodes (M10000).




(a)

(b)

Fig. 6. *Rawseeds* dataset, Bicocca outdoor: (a) odometric trajectory (solid line) with black triangles labeling the poses for which an absolute information from GPS is available, (b) estimated configuration (red) superimposed on the satellite map.

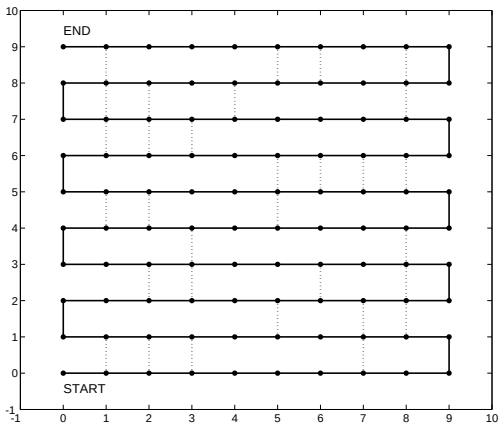



Fig. 7. Simulation scenario: the solid black line represents the odometric trajectory of the robot, travelling from the starting point $(0, 0)$ to the end $(0, 9)$. Nodes of the pose graph are denoted with black dots. The distance between two consecutive nodes is 1m. In this example the graph has 100 nodes. Loop closing constraints are shown as dotted lines. In this example we chose $\rho_{\text{lc}} = 0.5$, i.e., we add a loop closing from a node to one of its neighbours with probability 0.5.

results of the previous section: LAGO, g2o, g2o10, and iSAM attain the same objective value, while Toro is far from the optimal objective. For larger values of $\sigma_\delta$ we observe two trends. First, g2o10 outperforms g2o, meaning that (since the initial guess becomes inaccurate) performing more iterations improves the objective value. A similar observation holds for iSAM: since it does not stop after a fixed number of iterations, it is more accurate than g2o and g2o10. Second, for larger orientation uncertainties the problem of local minima becomes severe (as the initial guess becomes very imprecise). Therefore, Toro, which is known to have better convergence properties [24] becomes more accurate than all other techniques. Surprisingly, the objective attained by LAGO is practically the same as the one of iSAM. Figure 8(b) focuses on the comparison between LAGO and iSAM, and reveals that the accuracy of LAGO slightly degrades when increasing $\sigma_\delta$. This is consistent with our prediction of Section IV-D and Proposition 5. However, the performance degradation is practically negligible with respect to the performance loss of g2o and g2o10. Moreover, we notice that the quality of LAGO worsens only for $\sigma_\delta \geq 0.1\text{rad}$, which is already a large noise for common applications. CPU times of the compared approaches are shown in Figure 9(a). The plot shows that Toro, g2o, and g2o10 requires the same CPU time independently on the uncertainty, as they perform a fixed number of iterations. The computational effort of iSAM, instead, increases for larger $\sigma_\delta$, as the approach tends to perform more iterations. The computational cost of LAGO is independent of $\sigma_\delta$ and is a fraction of the other approaches. LAGO is able to attain the same accuracy of iSAM, while, for large orientation noise, iSAM requires a CPU time which is three orders of magnitude larger than LAGO.

*Influence of position measurement uncertainty*: A second test regards the performance of the compared approaches when varying the position measurement uncertainty $\sigma_\Delta$. We

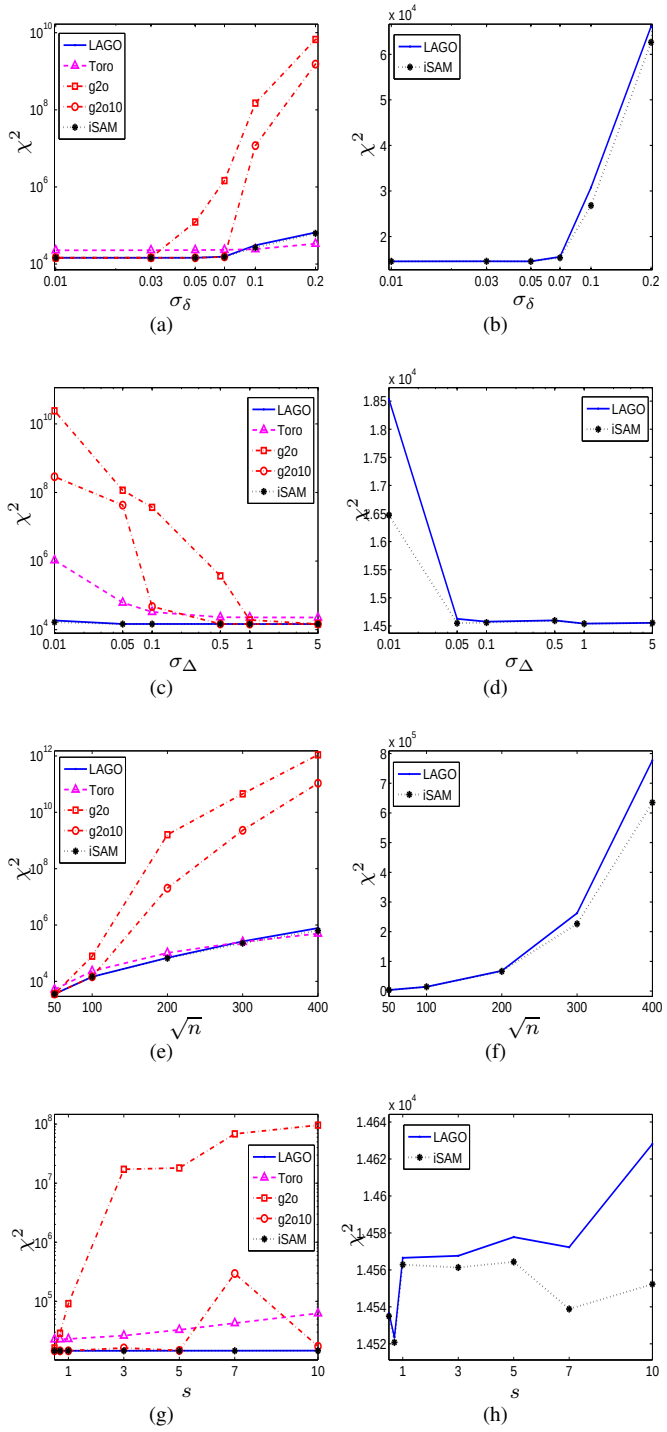



Fig. 8. Accuracy of LAGO, Toro, g2o, g2o10, and iSAM (smaller $\chi^2$ value means higher accuracy). (a-b) $\chi^2$ for different values of the orientation measurement uncertainty $\sigma_\delta$; (c-d) $\chi^2$ for different values of the position measurement uncertainty $\sigma_\Delta$; (e-f) $\chi^2$ for pose graphs with different number of nodes $n$; (g-h) $\chi^2$ for pose graphs with different inter-nodal distances $s$.

tested the following values of position uncertainty $\sigma_\Delta = \{0.01, 0.05, 0.1, 0.5, 1, 5\}\text{m}$, and we keep fixed the orientation uncertainty $\sigma_\delta = 0.05\text{rad}$. Figure 8(c) shows the accuracy of the five compared approaches, while Figure 8(d) focuses on

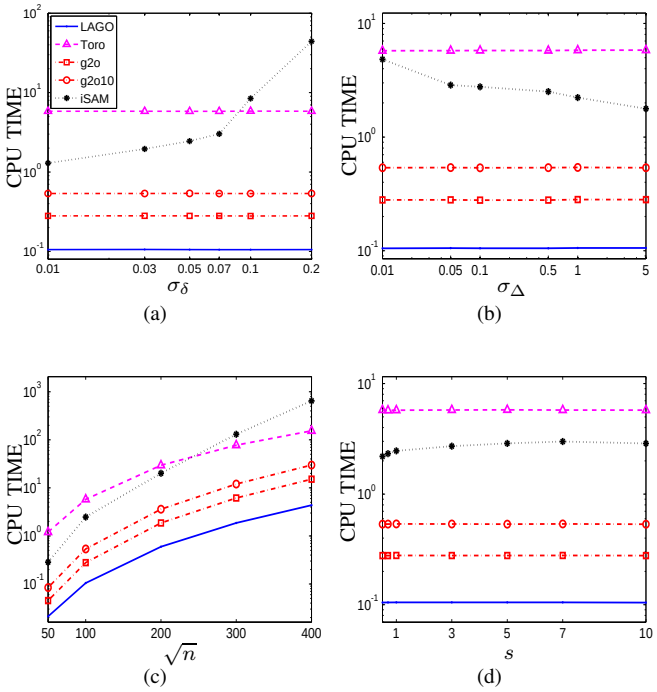



Fig. 9. Computational effort of LAGO, Toro, g2o, g2o10, and iSAM. (a) CPU times for different values of the orientation measurement uncertainty $\sigma_\delta$; (b) CPU times for different values of the position measurement uncertainty $\sigma_\Delta$; (c) CPU times for pose graphs with different number of nodes $n$; (d) CPU times for pose graphs with different inter-nodal distances $s$.

the comparison between LAGO and iSAM. For large values of $\sigma_\Delta$ we have that the solution of g2o, g2o10, and iSAM are very close, while Toro is less accurate (as for the case of small $\sigma_\delta$). Decreasing $\sigma_\Delta$ has the same effect of increasing $\sigma_\delta$: the approaches that perform more iterations (e.g., iSAM) outperform g2o and g2o. Moreover, for small values of $\sigma_\Delta$, Toro outperforms g2o and g2o10. An explanation for this phenomenon is that for small values of $\sigma_\Delta$ the non-convexities of the optimization problems becomes more relevant [6], which implies that iterative approaches requires more iterations and there is higher risk of being trapped in local minima. Figure 8(d) confirms the prediction of Proposition 5 (smaller $\sigma_\Delta$ corresponds to larger suboptimality gap); also in this case LAGO is accurate in common noise setups ($\sigma_\Delta > 0.1\text{m}$) and its performance degrades gracefully for small values of $\sigma_\Delta$. Figure 9(b) shows that LAGO implies a relevant computational advantage over the compared techniques in all tests.

*Influence of number of nodes*: We tested the performance of the approaches for different values of $n$ (number of nodes). We fixed $\sigma_\Delta = 0.5\text{m}$, $\sigma_\delta = 0.05\text{rad}$, $\rho_{\text{lc}} = 0.5$, and we built different pose graphs with $n = \{2500, 10000, 40000, 90000, 160000\}$ (for readability, in the figures we use the quantity $\sqrt{n} = \{50, 100, 200, 300, 400\}$). Results are shown in Figures 8(e)-(f). For large number of nodes the performance of LAGO, iSAM, and Toro is comparable, while the number of iterations in g2o and g2o10 are not sufficient to attain a small cost function. Figure 8(f) shows that the accuracy of LAGO degrades for very large pose graphs. This was already predictable from Proposition 5: a larger number of nodes implies a larger number of edges; this, in turns, leads to larger values of $\|\mathbf{\Delta}^l\|$, which contribute to increase the suboptimality gap (34). Another interesting result is the one of Figure 9(c). The CPU times of Toro and iSAM are already prohibitive for relatively small problem instances (90000 nodes). The CPU time of g2o remains below 20 seconds (this value is attained in the problem with 160000 nodes). LAGO in all instances is the fastest technique, assuring a speed-up of orders of magnitude with respect to iSAM and Toro. LAGO is able to solve the problem with 160000 nodes in less than 5 seconds. The dependence of the CPU time on the number of nodes is linear for LAGO (the plot is in logarithmic scale) and this is a desirable feature for large-scale SLAM.

*Influence of inter-nodal distances*: A last set of experiments evaluates the performance of the approaches for different inter-nodal distances. This kind of evaluation is less common in related work; however, according to the theoretical prediction of Proposition 5, we expect inter-nodal distances to influence the performance of LAGO: in fact, the term $\|\mathbf{\Delta}^l\|^2$ appearing in (34) is nothing else than the sum of the squared distances between pairs of nodes in the pose graph. In order to test the performance of the compared approaches for different inter-nodal distances, we consider a pose graph with $10^4$ nodes. Then, we "scale" the pose graph by multiplying the relative position measurements by a factor $s$. For $s = 1$ we retrieve the original graph (where we set the relative distance between nodes to 1m), while for $s > 1$ we obtain a stretched version of the graph (in which the relative distances between nodes is $s$). In our experiments we fix $\sigma_\Delta = 0.5\text{m}$, $\sigma_\delta = 0.05\text{rad}$, $\rho_{\text{lc}} = 0.5$, and we build different pose graphs with $s = \{0.5, 0.7, 1, 3, 5, 7, 10\}$. The accuracy of the compared techniques is shown in Figures 8(g)-(h). Consistently with our theoretical prediction, the accuracy of LAGO worsens for large values of $s$ (which lead to large values of $\|\mathbf{\Delta}^l\|$). The interesting thing is that also the accuracy of g2o and Toro degrades when increasing $s$. This may appear surprising, however, the recent work [6] shows that the non-convexities of pose graph optimization also increase for large inter-nodal distances, therefore, as in the previous cases, iterative techniques may require more iterations for problems that are far from being convex. The CPU times of the approaches are reported in Figure 9(d).

We also tested the five algorithms varying the number of loop closings in the graph, through the parameter $\rho_{\text{lc}}$. The approaches did not show remarkable influence on this parameter (besides the obvious increase in the cost due to the addition of edges) and we omit the results for space reasons.

*Summary of the results*: The results of this section can be summarized as follows. In most tests, Toro is less accurate than other state-of-the-art approaches and it tends to be slower. iSAM assures the best accuracy, although, for very noisy scenarios, Toro is able to attain smaller $\chi^2$ values, as it is less prone to local minima. Despite its accuracy, iSAM becomes fairly slow for relatively small problem instances ($> 10000$ nodes), or when the initial guess is poor (as iSAM requires several iterations to converge). g2o and g2o10 scale better in the number

of nodes and can perform optimization in few tens of seconds in the large-scale problem with 160000 nodes. However, these approaches tend to be less accurate than iSAM as they perform a fixed number of iterations. The proposed approach, LAGO, required the smallest computational effort with respect to all compared approaches in all tests. The objective value attained by LAGO is practically indistinguishable from iSAM. Slight performance degradation is observed in four cases: large orientation noise, small position noise, very large number of nodes, and large inter-nodal distances. Luckily, the accuracy of LAGO degrades gracefully and the degradation occurs for input data that are uncommon in the SLAM setup (e.g., very large orientation noise or very large inter-nodal distances). Despite its accuracy, LAGO is orders of magnitude faster than iSAM and is able to solve large-scale problems in few seconds, entailing a remarkable practical advantage in planar problems.

## VII. Conclusion

We presented an approximation for pose graph optimization, under the assumption that the relative position and the relative orientation measurements are independent. The approximation, named LAGO (*Linear Approximation for pose Graph Optimization*), exploits two properties of the optimization problem; first, orientation estimation can be framed in terms of linear estimation in a planar problem; second, given robot orientations, the pose estimation problem also becomes a linear estimation problem. LAGO requires no initial guess for optimization and can be computed in closed form. We provide an analytical assessment of the approximation, identifying the factors influencing its quality. LAGO can be used as a linear initialization for iterative techniques or as a stand-alone tool. Empirical evidence suggests that the approximation is very accurate in practice, and, in common SLAM scenarios, it is able to attain the same minimum of iterative techniques at a fraction of the computational cost.

While in this work we mainly pointed out the advantages of LAGO in terms of accuracy and computational effort, current research is devoted to robustness issues: the insights underlying LAGO provide several tools to enhance global convergence of pose graph optimization. A recent example of this effort is the recent preprint [11], where a multi-hypothesis algorithm is proposed for solving an orientation-only estimation problem. Future work also includes extensions of LAGO to 3D scenarios. This extension is nontrivial as the first phase of LAGO can no longer be formulated in a linear fashion in non-planar scenarios.

## Appendix A

In this appendix we prove the result of Proposition 3, showing that the two linear systems to be solved in LAGO (in Phase 1 and Phase 3) admit a unique solution. Let us start from Phase 1. We already observed in Lemma 1 that the reduced incidence matrix $\boldsymbol{A}$ is full column rank. The information matrix $\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}$ is positive-definite by definition. Therefore, $\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}\boldsymbol{A}^{\mathsf{T}}$ is positive-definite, hence invertible [27], and the orientation estimate $\hat{\boldsymbol{\theta}} = (\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}\boldsymbol{\delta}$ is unique. For demonstrating the uniqueness of the solution of the linear system of Phase 3, we first have to show that the matrix $\boldsymbol{P}_{\boldsymbol{z}}$, computed in the Phase 2 is positive-definite. For this purpose we observe that the matrix $\boldsymbol{P}_{\boldsymbol{\Delta}^l}$ is positive-definite by definition. Therefore the matrix $\mathrm{diag}(\boldsymbol{P}_{\boldsymbol{\Delta}^l}, \boldsymbol{P}_{\hat{\boldsymbol{\theta}}})$ is also positive-definite. Because of the block structure of $\boldsymbol{H}$, it follows that $\det(\boldsymbol{H}) = \det(\hat{\boldsymbol{R}})\det(\mathbf{I}_n) = \det(\hat{\boldsymbol{R}})$ [27]. $\hat{\boldsymbol{R}}$ is a block diagonal matrix, whose 2 by 2 blocks are rotation matrix, hence $\det(\hat{\boldsymbol{R}}) = \prod_{(i,j)\in\mathcal{E}} \det(\boldsymbol{R}_i) = 1$. Therefore $\det(\boldsymbol{H}) = 1$ and $\boldsymbol{H}$ is full rank. Accordingly, the matrix $\boldsymbol{H}\ \mathrm{diag}(\boldsymbol{P}_{\boldsymbol{\Delta}^l}, \boldsymbol{P}_{\hat{\boldsymbol{\theta}}})\ \boldsymbol{H}^{\mathsf{T}}$ is positive-definite [27]. Because of the structure of matrix $\boldsymbol{B}$, it holds $\mathrm{rank}(\boldsymbol{B}) = \mathrm{rank}(\boldsymbol{A}_2) + \mathrm{rank}(\mathbf{I}_n) = \mathrm{rank}(\boldsymbol{A}_2) + n$. The rank of the reduced incidence matrix $\boldsymbol{A}$ is $n$, by Lemma 1. Therefore, $\mathrm{rank}(\boldsymbol{A}_2) = \mathrm{rank}(\boldsymbol{A} \otimes \mathbf{I}_2) = 2n$, and it follows that $\mathrm{rank}(\boldsymbol{B}) = 3n$. Since $\boldsymbol{P}_{\boldsymbol{z}}^{-1}$ is positive-definite and $\boldsymbol{B}$ is full rank, then $\boldsymbol{B}\boldsymbol{P}_{\boldsymbol{z}}^{-1}\boldsymbol{B}^{\mathsf{T}}$ is positive-definite, hence invertible [27], from which the uniqueness claim follows. □

## Appendix B

In this appendix we prove the claim of Proposition 4. We have already shown in Section IV-C that the approximation $\boldsymbol{x}^\star$ is composed by the suboptimal solution $\hat{\boldsymbol{x}}$ plus a correction term $\tilde{\boldsymbol{x}}$. Now the demonstration reduces to verify that $\tilde{\boldsymbol{x}} = [\tilde{\boldsymbol{p}}^{\mathsf{T}}\ \tilde{\boldsymbol{\theta}}^{\mathsf{T}}]^{\mathsf{T}}$ corresponds to a Gauss-Newton correction of our optimization problem (10) around $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{p}}^{\mathsf{T}}\ \hat{\boldsymbol{\theta}}^{\mathsf{T}}]^{\mathsf{T}}$. In order to prove this point we compute a first-order approximation of the residual errors in (10) around the suboptimal solution $\hat{\boldsymbol{x}}$:

$$\begin{aligned} f(\boldsymbol{x}) \approx &\left(\boldsymbol{A}_2^{\mathsf{T}}\hat{\boldsymbol{p}} + \boldsymbol{A}_2^{\mathsf{T}}\boldsymbol{\eta}_{\boldsymbol{p}} - \hat{\boldsymbol{R}}\boldsymbol{\Delta}^l - \boldsymbol{J}\boldsymbol{\eta}_{\boldsymbol{\theta}}\right)^{\mathsf{T}} \\ &\left(\hat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{\Delta}^l}\hat{\boldsymbol{R}}^{\mathsf{T}}\right)^{-1}\left(\boldsymbol{A}_2^{\mathsf{T}}\hat{\boldsymbol{p}} + \boldsymbol{A}_2^{\mathsf{T}}\boldsymbol{\eta}_{\boldsymbol{p}} - \hat{\boldsymbol{R}}\boldsymbol{\Delta}^l - \boldsymbol{J}\boldsymbol{\eta}_{\boldsymbol{\theta}}\right) + \\ &+\left(\boldsymbol{A}^{\mathsf{T}}\hat{\boldsymbol{\theta}} + \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\eta}_{\boldsymbol{\theta}} - \boldsymbol{\delta}\right)^{\mathsf{T}}\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}\left(\boldsymbol{A}^{\mathsf{T}}\hat{\boldsymbol{\theta}} + \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\eta}_{\boldsymbol{\theta}} - \boldsymbol{\delta}\right), \end{aligned} \tag{48}$$

where $\boldsymbol{\eta}_{\boldsymbol{\theta}}$ and $\boldsymbol{\eta}_{\boldsymbol{p}}$ are the displacements from the linearization point and the covariance matrix $\boldsymbol{R}\boldsymbol{P}_{\boldsymbol{\Delta}^l}\boldsymbol{R}^{\mathsf{T}}$ is evaluated in $\hat{\boldsymbol{\theta}}$. This convex approximation can be minimized by taking the first derivative with respect to the optimization variables $\boldsymbol{\eta}_{\boldsymbol{\theta}}$ and $\boldsymbol{\eta}_{\boldsymbol{p}}$ and imposing it to be zero:

$$\begin{cases} \frac{\partial f}{\partial \boldsymbol{\eta}_{\boldsymbol{p}}} = 2\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\left(\boldsymbol{A}_2^{\mathsf{T}}\hat{\boldsymbol{p}} + \boldsymbol{A}_2^{\mathsf{T}}\boldsymbol{\eta}_{\boldsymbol{p}} - \hat{\boldsymbol{R}}\boldsymbol{\Delta}^l - \boldsymbol{J}\boldsymbol{\eta}_{\boldsymbol{\theta}}\right) = \mathbf{0}_{2n} \\ \frac{\partial f}{\partial \boldsymbol{\eta}_{\boldsymbol{\theta}}} = 2\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{\delta}}^{-1}\left(\boldsymbol{A}^{\mathsf{T}}\hat{\boldsymbol{\theta}} + \boldsymbol{A}^{\mathsf{T}}\boldsymbol{\eta}_{\boldsymbol{\theta}} - \boldsymbol{\delta}\right) + \\ -2\boldsymbol{J}^{\mathsf{T}}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\left(\boldsymbol{A}_2^{\mathsf{T}}\hat{\boldsymbol{p}} + \boldsymbol{A}_2^{\mathsf{T}}\boldsymbol{\eta}_{\boldsymbol{p}} - \hat{\boldsymbol{R}}\boldsymbol{\Delta}^l - \boldsymbol{J}\boldsymbol{\eta}_{\boldsymbol{\theta}}\right) = \mathbf{0}_n \end{cases}, \tag{49}$$

where $\boldsymbol{P}_{\boldsymbol{\Delta}^g} = \hat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{\Delta}^l}\hat{\boldsymbol{R}}^{\mathsf{T}}$. From the first equation, substituting the expression of $\hat{\boldsymbol{p}}$, given in (21), we can rewrite $\boldsymbol{\eta}_{\boldsymbol{p}}$ in function of $\boldsymbol{\eta}_{\boldsymbol{\theta}}$:

$$\boldsymbol{\eta}_{\boldsymbol{p}} = \boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J}\boldsymbol{\eta}_{\boldsymbol{\theta}}. \tag{50}$$

Substituting the expressions (19) and (50) in the second equation of (49) we can obtain the value of $\boldsymbol{\eta}_{\boldsymbol{\theta}}$ which annihilates the first derivative of the cost function (48):

$$\boldsymbol{\eta}_{\boldsymbol{\theta}} = \boldsymbol{P}_{\boldsymbol{\theta}^\star}\boldsymbol{J}^{\mathsf{T}}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\left(\boldsymbol{A}_2^{\mathsf{T}}\boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1} - \mathbf{I}_{2m}\right)\hat{\boldsymbol{R}}\boldsymbol{\Delta}^l \tag{51}$$

with $\boldsymbol{P}_{\boldsymbol{\theta}^\star}$ defined as in (28). It is now easy to see that the previous expression coincides with $\tilde{\boldsymbol{\theta}}$ in (31). Finally if we substitute again (51) in (50) we can compute

$$\boldsymbol{\eta}_{\boldsymbol{p}} = \boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\boldsymbol{J}\boldsymbol{P}_{\boldsymbol{\theta}^\star}\boldsymbol{J}^{\mathsf{T}}\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1}\left(\boldsymbol{A}_2^{\mathsf{T}}\boldsymbol{P}_{\hat{\boldsymbol{p}}}\boldsymbol{A}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}^{-1} - \mathbf{I}_{2m}\right)\hat{\boldsymbol{R}}\boldsymbol{\Delta}^l, \tag{52}$$

which can be seen to coincide with $\tilde{\boldsymbol{p}}$, see (31), proving our thesis. □

## APPENDIX C

In this appendix we show that the suboptimal orientation estimate $\hat{\boldsymbol{\theta}}$ and (if needed) the suboptimal position estimate $\hat{\boldsymbol{p}}$ can be computed with a complexity that depends on the number of loop closings in the pose graph. We assume that the relative orientation measurements have already been regularized. Let us define the vector $\hat{\boldsymbol{\delta}} \doteq \boldsymbol{A}^{\mathsf{T}}\hat{\boldsymbol{\theta}}$. Note that if we are able to compute $\hat{\boldsymbol{\delta}}$, we can easily obtain $\hat{\boldsymbol{\theta}}$; for instance, consider the graph in Figure 1: if $\hat{\boldsymbol{\delta}} = [\hat{\delta}_{01}\ \hat{\delta}_{12}\ \hat{\delta}_{23}\ \hat{\delta}_{34}\ \hat{\delta}_{41}]^{\mathsf{T}}$ is known, then we can compute the suboptimal nodes orientations as $\hat{\theta}_1 = \hat{\delta}_{01}$, $\hat{\theta}_2 = \hat{\delta}_{01} + \hat{\delta}_{12}$, $\hat{\theta}_3 = \hat{\delta}_{01} + \hat{\delta}_{12} + \hat{\delta}_{23}$, and $\hat{\theta}_4 = \hat{\delta}_{01} + \hat{\delta}_{12} + \hat{\delta}_{23} + \hat{\delta}_{34}$. In general, if we order $\hat{\boldsymbol{\delta}}$ as $\hat{\boldsymbol{\delta}} = [\hat{\boldsymbol{\delta}}_{\text{odo}}^{\mathsf{T}}\ \hat{\boldsymbol{\delta}}_{\text{lc}}^{\mathsf{T}}]$ such that $\hat{\boldsymbol{\delta}}_{\text{odo}}$ corresponds to the edges of the (odometric) spanning tree (i.e., $\hat{\boldsymbol{\delta}}_{\text{odo}} = [\hat{\boldsymbol{\delta}}_{01}\ \hat{\boldsymbol{\delta}}_{12}\ \dots\ \hat{\boldsymbol{\delta}}_{(n-1,n)}]^{\mathsf{T}}$), and $\hat{\boldsymbol{\delta}}_{\text{lc}}$ are the corresponding chords, then we can compute $\hat{\boldsymbol{\theta}} = \boldsymbol{V}\hat{\boldsymbol{\delta}}$, with $\boldsymbol{V} = [\boldsymbol{U}_{n\times n}\ \boldsymbol{0}_{n\times m-n}]$ and $\boldsymbol{U}_{n\times n}$ is a lower triangular matrix of size $n$ with entries below the main diagonal (an on the main diagonal itself) equal to 1. Then, the following fact holds.

**Proposition 14.** *Given the regularized orientation measurements $\boldsymbol{\delta}$ and the corresponding covariance matrix $\boldsymbol{P}_{\boldsymbol{\delta}}$, it is possible to compute the suboptimal orientation estimate $\hat{\boldsymbol{\theta}}$ as $\hat{\boldsymbol{\theta}} = \boldsymbol{V}\hat{\boldsymbol{\delta}}$ with:*

$$\hat{\boldsymbol{\delta}} = \begin{cases} \boldsymbol{\delta} & \text{if } m = n \\ \boldsymbol{\delta} - \boldsymbol{P}_{\boldsymbol{\delta}}\boldsymbol{C}^{\mathsf{T}}(\boldsymbol{C}\boldsymbol{P}_{\boldsymbol{\delta}}\boldsymbol{C}^{\mathsf{T}})^{-1}\boldsymbol{C}\boldsymbol{\delta} & \text{if } m > n \end{cases}, \tag{53}$$

*where $\boldsymbol{C}$ is a cycle basis matrix of the pose graph.*

**Proof.** The proof is a straightforward consequence of Theorem 1 in [47], and of the equality $\hat{\boldsymbol{\theta}} = \boldsymbol{V}\hat{\boldsymbol{\delta}}$. □

Although we already observed that the approach proposed in this article does not require the computation of $\hat{\boldsymbol{p}}$, we further generalize the previous result to the suboptimal position estimate. The proof of the following result follows from [47].

**Proposition 15.** *Given the suboptimal orientation estimate $\hat{\boldsymbol{\theta}}$, the relative position measurements $\boldsymbol{\Delta}^l$, and the corresponding covariance matrix $\boldsymbol{P}_{\boldsymbol{\Delta}^l}$, the suboptimal position estimate $\hat{\boldsymbol{p}}$ in* (21)*, can be alternatively computed as $\hat{\boldsymbol{p}} = \boldsymbol{V}_2\hat{\boldsymbol{\Delta}}^g$, with:*

$$\hat{\boldsymbol{\Delta}}^g = \begin{cases} \hat{\boldsymbol{R}}\boldsymbol{\Delta}^l & \text{if } m = n \\ \hat{\boldsymbol{R}}\boldsymbol{\Delta}^l - \boldsymbol{P}_{\boldsymbol{\Delta}^g}\boldsymbol{C}_2^{\mathsf{T}}(\boldsymbol{C}_2\boldsymbol{P}_{\boldsymbol{\Delta}^g}\boldsymbol{C}_2^{\mathsf{T}})^{-1}\boldsymbol{C}_2\hat{\boldsymbol{R}}\boldsymbol{\Delta}^l & \text{if } m > n \end{cases}, \tag{54}$$

*where $\boldsymbol{C}_2 = \boldsymbol{C} \otimes \mathbf{I}_2$, $\boldsymbol{V}_2 = \boldsymbol{V} \otimes \mathbf{I}_2$, $\hat{\boldsymbol{R}}$ is built as described in Section IV, and $\boldsymbol{P}_{\boldsymbol{\Delta}^g} \doteq \hat{\boldsymbol{R}}\boldsymbol{P}_{\boldsymbol{\Delta}^l}\hat{\boldsymbol{R}}^{\mathsf{T}}$.* □

## APPENDIX D

This appendix contains a technical lemma, useful in the proof of Proposition 5 (given in Appendix E). The first claim is a well known relation from graph theory [13]. The second claim follows from the first and from equation (5.13.6) in [41].

**Lemma 16** (Orthogonal complements and projectors [47])**.** *Let us consider a cycle basis matrix $\boldsymbol{C} \in \mathbb{Z}^{\ell\times m}$ and the reduced incidence matrix $\boldsymbol{A} \in \mathbb{Z}^{n\times m}$ of a connected graph $\mathcal{G}$. Then the following facts hold true:*

1) *$\boldsymbol{C}^{\mathsf{T}}$ and $\boldsymbol{A}^{\mathsf{T}}$ are orthogonal complements, i.e., $[\boldsymbol{A}^{\mathsf{T}}\ \boldsymbol{C}^{\mathsf{T}}]$ is a square matrix of full rank and $\boldsymbol{C}\boldsymbol{A}^{\mathsf{T}} = \boldsymbol{0}_{\ell\times n}$;*
2) *$\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{A} + \boldsymbol{C}^{\mathsf{T}}(\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}})^{-1}\boldsymbol{C} = \mathbf{I}_m$.* □

## APPENDIX E

This appendix contains the proof of Proposition 5. We have to prove that $\beta \doteq \hat{f}_n - f_n^{\text{ML}} \leq \frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\|\boldsymbol{\Delta}^l\|^2$. Let us write explicitly the quantity $\beta \doteq \hat{f}_n - f_n^{\text{ML}}$, recalling that $\hat{f}_n$ and $f_n^{\text{ML}}$ can be obtained by substituting $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}^{\text{ML}}$ in the normalized cost (33), respectively:

$$\beta = \frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\left\|\boldsymbol{A}_2^{\mathsf{T}}\hat{\boldsymbol{p}} - \boldsymbol{R}(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta}^l\right\|^2 + \left\|\boldsymbol{A}^{\mathsf{T}}\hat{\boldsymbol{\theta}} - \boldsymbol{\delta}\right\|^2 + \tag{55}$$

$$-\frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\left\|\boldsymbol{A}_2^{\mathsf{T}}\boldsymbol{p}^{\text{ML}} - \boldsymbol{R}(\theta^{\text{ML}})\boldsymbol{\Delta}^l\right\|^2 - \left\|\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\theta}^{\text{ML}} - \boldsymbol{\delta}\right\|^2. \tag{56}$$

The vector $\hat{\boldsymbol{\theta}}$ attains the minimum of $\left\|\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\theta} - \boldsymbol{\delta}\right\|^2$, as per equation (19), therefore it holds $\left\|\boldsymbol{A}^{\mathsf{T}}\hat{\boldsymbol{\theta}} - \boldsymbol{\delta}\right\|^2 - \left\|\boldsymbol{A}^{\mathsf{T}}\boldsymbol{\theta}^{\text{ML}} - \boldsymbol{\delta}\right\|^2 \leq 0$, for any $\boldsymbol{\theta}^{\text{ML}}$. Moreover, the quantity $\frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\left\|\boldsymbol{A}_2^{\mathsf{T}}\boldsymbol{p}^{\text{ML}} - \boldsymbol{R}(\theta^{\text{ML}})\boldsymbol{\Delta}^l\right\|^2$ is always non-negative, therefore it holds

$$\beta \leq \frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\left\|\boldsymbol{A}_2^{\mathsf{T}}\hat{\boldsymbol{p}} - \boldsymbol{R}(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta}^l\right\|^2. \tag{57}$$

Under the assumption $\boldsymbol{P}_{\boldsymbol{\Delta}^l} = \sigma_{\boldsymbol{\Delta}}^2\mathbf{I}_{2m}$, the expression of $\hat{\boldsymbol{p}}$ given in (21) becomes:

$$\hat{\boldsymbol{p}} = \left(\boldsymbol{A}_2\boldsymbol{A}_2^{\mathsf{T}}\right)^{-1}\boldsymbol{A}_2\boldsymbol{R}(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta}^l. \tag{58}$$

Substituting the previous expressions in (57) we obtain:

$$\beta \leq \frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\left\|\left(\boldsymbol{A}_2^{\mathsf{T}}\left(\boldsymbol{A}_2\boldsymbol{A}_2^{\mathsf{T}}\right)^{-1}\boldsymbol{A}_2 - \mathbf{I}_{2m}\right)\boldsymbol{R}(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta}^l\right\|^2. \tag{59}$$

Recalling that $\boldsymbol{A}_2 = \boldsymbol{A} \otimes \mathbf{I}_2$ and $\mathbf{I}_{2m} = \mathbf{I}_m \otimes \mathbf{I}_2$, and using Lemma 16, we rewrite (59) as:

$$\beta \leq \frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\left\|\left[\left(\boldsymbol{A}^{\mathsf{T}}\left(\boldsymbol{A}\boldsymbol{A}^{\mathsf{T}}\right)^{-1}\boldsymbol{A} - \mathbf{I}_m\right) \otimes \mathbf{I}_2\right]\boldsymbol{R}(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta}^l\right\|^2 =$$

$$= \frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\left\|\left[\left(\boldsymbol{C}^{\mathsf{T}}\left(\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}\right)^{-1}\boldsymbol{C}\right) \otimes \mathbf{I}_2\right]\boldsymbol{R}(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta}^l\right\|^2. \tag{60}$$

Using standard norm inequalities we obtain

$$\beta \leq \frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\left\|\left(\boldsymbol{C}^{\mathsf{T}}\left(\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}\right)^{-1}\boldsymbol{C}\right) \otimes \mathbf{I}_2\right\|^2\left\|\boldsymbol{R}(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta}^l\right\|^2 =$$

$$= \frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\left\|\boldsymbol{C}^{\mathsf{T}}\left(\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}\right)^{-1}\boldsymbol{C}\right\|^2\left\|\boldsymbol{R}(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta}^l\right\|^2. \tag{61}$$

We now observe that $\boldsymbol{C}^{\mathsf{T}}\left(\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}\right)^{-1}\boldsymbol{C}$ is an orthogonal projection matrix, since it is idempotent and symmetric, therefore $\|\boldsymbol{C}^{\mathsf{T}}\left(\boldsymbol{C}\boldsymbol{C}^{\mathsf{T}}\right)^{-1}\boldsymbol{C}\| = 1$, see equation (5.13.10) in [41]. Therefore, from (61), it holds $\beta \leq \frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\|\boldsymbol{R}(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta}^l\|$. Finally, we notice that $\boldsymbol{R}(\hat{\boldsymbol{\theta}})$ is a rotation matrix, hence $\|\boldsymbol{R}(\hat{\boldsymbol{\theta}})\boldsymbol{\Delta}^l\| = \|\boldsymbol{\Delta}^l\|$, which leads to the desired upper bound $\beta \leq \frac{\sigma_{\boldsymbol{\delta}}^2}{\sigma_{\boldsymbol{\Delta}}^2}\|\boldsymbol{\Delta}^l\|^2$. □

## REFERENCES


[1] P. Barooah. Estimation and control with relative measurements: Algorithms and scaling laws. *Ph.D. Thesis*, University of California, Santa Barbara, 2007.

[2] P. Barooah and J.P. Hespanha. Estimation on graphs from relative measurements. *Control Systems Magazine*, 27(4):57–74, 2007.

[3] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D.G. Sorrenti, and J.D. Tardós. RAWSEEDS: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets, http://www.rawseeds.org/. In *Proc. of the IROS'06 Workshop on Benchmarks in Robotics Research*, 2006.

[4] M. Bosse and R. Zlot. Keypoint design and evaluation for place recognition in 2d lidar maps. *Robotics and Autonomous System J.*, 57(12):1211–1224, 2009.

[5] C. Cadena, D. Galvez-Lopez, F. Ramos, J.D. Tardós, and J. Neira. Robust place recognition with stereo cameras. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent RObots and Systems (IROS)*, pages 5182–5189, 2010.

[6] L. Carlone. Convergence analysis of pose graph optimization via Gauss-Newton methods. accepted for publication at the Int. Conf. on Robotics and Automation (ICRA), available at: http://www.lucacarlone.com/index.php/publications, 2013.

[7] L. Carlone, R. Aragues, J.A. Castellanos, and B. Bona. A fast and accurate approximation for pose graph optimization. supplementary material, http://www.lucacarlone.com/index.php/resources/research/lago, 2013.

[8] L. Carlone, R. Aragues, J.A. Castellanos, and B. Bona. A first-order solution to simultaneous localization and mapping with graphical models. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1764–1771, 2011.

[9] L. Carlone, R. Aragues, J.A. Castellanos, and B. Bona. A linear approximation for graph-based simultaneous localization and mapping. In *Robotics: Science and Systems (RSS)*, 2011.

[10] L. Carlone, R. Aragues, J.A. Castellanos, and B. Bona. A linear approximation for nonlinear pose graph optimization. *technical report, available at: http://www.lucacarlone.com/index.php/resources/research/lago*, 2012.

[11] L. Carlone and A. Censi. From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization. ArXiv preprint: http://arxiv.org/abs/1211.3063, 2012.

[12] L. Carlone, J. Yin, S. Rosa, and Y. Zehui. Graph optimization with unstructured covariance: Fast, accurate, linear approximation. In *International Conference on simulation, modeling, and programming, for autonomous robots*, 2012.

[13] W. Chen. *Graph Theory and Its Engineering Applications*. Advanced Series in Electrical and Computer Engineering, 1997.

[14] D. Crandall, A. Ovens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3001–3008, 2011.

[15] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C. Thorpe. Subgraph-preconditioned conjugate gradients for large scale SLAM. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent RObots and Systems (IROS)*, pages 2566–2571, 2010.

[16] F. Dellaert and A. Stroupe. Linear 2D localization and mapping for single and multiple robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 688–694, 2002.

[17] G. Dubbelman, I. Esteban, and K. Schutte. Efficient trajectory bending with applications to loop closure. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent RObots and Systems (IROS)*, pages 1–7, 2010.

[18] G. Dubbelman, P. Hansen, B. Browning, and M.B. Dias. Orientation only loop-closing with closed-form trajectory bending. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 815–82, 2012.

[19] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): Part I. *Robotics and Automation Magazine*, 13:99–110, 2006.

[20] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): Part II. *Robotics and Automation Magazine*, 13:108–117, 2006.

[21] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Trans. on Robotics*, 21(2):196–207, 2005.

[22] V.M Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 684–691, 2004.

[23] G. Grisetti, R. Kuemmerle, C. Stachniss, U. Frese, and C. Hertzberg. A hierarchical optimization on manifolds for online 2D and 3D mapping. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 273–278, 2010.

[24] G. Grisetti, C. Stachniss, and W. Burgard. Non-linear constraint network optimization for efficient map learning. *IEEE Trans. on Intelligent Transportation Systems*, 10(3):428–439, 2009.

[25] J.L. Gross and T.W. Tucker. *Topological graph theory*. Wiley Interscience, 1987.

[26] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[27] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, UK, 1985.

[28] S. Huang, Y. Lai, U. Frese, and G. Dissanayake. How far is SLAM from a linear least squares problem? In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent RObots and Systems (IROS)*, pages 3011–3016, 2010.

[29] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3281–3288, 2011.

[30] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2:incremental smoothing and mapping using the bayes tree. *Int. J. Robot. Res.*, 31:217–236, 2012.

[31] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM:incremental smoothing and mapping. *IEEE Trans. on Robotics*, 24(6):1365–1378, 2008.

[32] T. Kavitha, C. Liebchen, K. Mehlhorn, D. Michail, R. Rizzi, T. Ueckerdt, and K. Zweig. Cycle bases in graphs: Characterization, algorithms, complexity, and applications. *Computer Science Rev.*, 3(4):199–243, 2009.

[33] K. Konolige. Large-scale map-making. In *Proc. of the AAAI National Conf. on Artificial Intelligence*, 2004.

[34] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent. Efficient sparse pose adjustment for 2D mapping. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent RObots and Systems (IROS)*, pages 22–29, 2010.

[35] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3607–3613, 2011.

[36] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner. On measuring the accuracy of SLAM algorithms. *Autonomous Robots*, 27(4):387–407, 2009.

[37] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner. Slam benchmarking webpage. *http://ais.informatik.uni-freiburg.de/slamevaluation*, 2009.

[38] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[39] D. Martinec and T. Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[40] J.M. Mendel. Lessons in estimation theory for signal processing, communications, and control. *Englewood Cliffs, NJ: Prentice-Hall*, 1995.

[41] C.D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

[42] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. In *Robotics: Science and Systems (RSS)*, 2012.

[43] Edwin Olson, John Leonard, and Seth Teller. Fast iterative alignment of pose graphs with poor estimates. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2262–2269, 2006.

[44] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *IEEE Trans. Pattern Anal. Machine Intell.*, 19:834–845, 1997.

[45] D. Lodi Rizzini. A closed-form constraint networks solver for maximum likelihood mapping. In *Proc. of the European Conference on Mobile Robots (ECMR)*, pages 223–228, 2010.

[46] D.M. Rosen, M. Kaess, and J.J. Leonard. An incremental trust-region method for robust online sparse least-squares estimation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1262–1269, 2012.

[47] W.J. Russell, D.J. Klein, and J.P. Hespanha. Optimal estimation on the graph cycle space. *IEEE Trans. On Signal Processing*, 59(6):2834–2846, 2011.

[48] D. Sabatta, D. Scaramuzza, and R. Siegwart. Improved appearance-based matching in similar and dynamic environments using a vocabulary tree. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2262–2269, 2010.

[49] G.C. Sharp, S.W. Lee, and D.K. Wehe. Multiview registration of 3d scenes by minimizing error between coordinate frames. In *Lecture Notes in Computer Science, Computer Vision - 7th European Conference on Computer Vision*, pages 587–597, 2002.

[50] C. Stachniss, U. Frese, and G. Grisetti. OpenSLAM, http://www.openslam.org/.

[51] N. Sünderhauf and P. Protzel. Switchable constraints for robust pose graph slam. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent RObots and Systems (IROS)*, 2012.

[52] N. Sünderhauf and P. Protzel. Towards a robust back-end for pose graph slam. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[53] S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *Int. J. Robot. Res.*, 25:403–429, 2006.

[54] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. *Bundle adjustment - a modern synthesis*. in Vision Algorithms: Theory and Practice, ser. LNCS, W. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer Verlag, 1999.

[55] H Wang, G Hu, S Huang, and G Dissanayake. On the Structure of Nonlinearities in Pose Graph SLAM. In *Robotics: Science and Systems (RSS)*, 2012.