A Linear Approximation for Pose Graph Optimization

Luca Carlone, *Student Member, IEEE,* Rosario Aragues, *Student Member, IEEE,*
José A. Castellanos, *Member, IEEE,* and Basilio Bona, *Member, IEEE*

Abstract

In this work we investigate the problem of *Simultaneous Localization And Mapping* (SLAM) for the case in which robot measurements are modeled as a network of constraints in a *pose graph*. We combine tools belonging to linear estimation and graph theory to devise a closed-form approximation to the *batch* SLAM problem, under the assumption that relative position and relative orientation measurements are independent. The approach needs no initial guess for optimization and is formally proven to admit solution under the SLAM setup. The resulting estimate can be used as an approximation of the actual nonlinear solution or can be further refined by using it as initial guess for nonlinear optimization techniques. Experimental analysis demonstrates that such refinement is often unnecessary, since the linear estimate is already accurate in practice. Furthermore, we discuss how the approach allows to mitigate the orientation wraparound problem which is known to prevent convergence in state-of-the-art techniques.

Index Terms

Pose graph optimization, Simultaneous Localization And Mapping, Graph Theory, Linear Estimation.

I. Introduction

In the landscape of mobile robots navigation it has been recognized the fundamental role played by Simultaneous Localization and Mapping for the deployment of truly autonomous agents [12]. When a robot travels in an unknown scenario, the construction of a world model and the concurrent estimation of robot location may be crucial for mission accomplishment, for enhancing motion planning effectiveness, and for attaining a desired level of situational awareness. As a consequence, since the first probabilistic statement of the problem, dating back to 1986, SLAM has been a central topic of research, this being witnessed by the large amount of contributions that can be found in literature.

Literature on SLAM can be sectioned along different directions. A possible classification of SLAM problems distinguishes between *online* and *batch* SLAM approaches. The former encompasses the techniques that incrementally estimate the SLAM posterior by recursively including the most recent measurements in the posterior density. This is typically the case of a robot that acquires proprioceptive and exteroceptive measurements from the environment and, at each time step, incrementally includes such information in the posterior describing robot pose and a representation of the surrounding environment. Batch SLAM approaches, instead, tackle the problem of retrieving the whole posterior, 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.

Graph formalism has been demonstrated to be a powerful tool for solving batch SLAM. In graph-based approaches, measurements acquired during robot motion are modeled as constraints in a graph of poses. The use of graph-based models in SLAM has several desirable properties, allowing, for instance, to easily

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 and DPI2009-13710, and grant MEC BES-2007-14772.

L. Carlone and B. Bona are with the Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino, Italy, {luca.carlone, basilio.bona}@polito.it

R. Aragues and J.A. Castellanos are with the Departamento de Informática e Ingeniería de Sistemas, Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Zaragoza, Spain, {raragues, jacaste}@unizar.es

introduce absolute position information (i.e., GPS) or to recover from bad loop closings. The seminal paper [26] paved the way for several results towards the objective of a sustainable solution to batch SLAM. The setup considered by Lu and Milios was based on scan matching for retrieving relative constraints between robot poses. This framework was extended to the general case of a graph containing both robot poses and landmark positions in [34]. Thrun and Montemerlo showed that it is possible to marginalize out variables corresponding to landmarks, hence reducing the problem to the pose estimation setup discussed in [26]. Moreover, they enabled the estimation of large maps, applying a conjugate gradient-based optimization. Konolige investigated a reduction scheme for the purpose of improving the computational effort of nonlinear optimization [23]: the optimization process was restricted to poses involved in at least one loop closing, providing a remarkable advantage for graphs with low connectivity. Frese *et al.* proposed a multilevel relaxation approach for batch SLAM [13], allowing to considerably reduce the computational time of optimization by applying a multi-grid algorithm. A further breakthrough in the literature of graph-based approaches came with the use of *incremental pose parametrization*, proposed in [28]. Olson *et al.* showed how a clever selection of the optimization variables can greatly simplify problem structure, enabling fast computation. Moreover they proposed not to optimize all the constraints at the same time, but to sequentially select each constraint and refine nodes' configuration so to reduce the residual error for such constraint. Grisetti *et al.* extended such framework, taking advantage of the use of stochastic gradient descent in planar and three-dimensional scenarios [14], [15]. Moreover they proposed a parametrization based on a spanning tree of the graph, for speeding up the correction of the residual errors over the network. All the aforementioned 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. In particular, when a linear approximation of the residual errors in the cost function is considered, the local problem becomes an unconstrained quadratic problem and the local correction can be obtained as solution of a system of linear equations. In such a case the source of complexity stems from the need of repeatedly solving a large-scale linear system. A discussion on the performances of *direct* and *iterative* methods for solving linear systems, with application to SLAM, can be found in [10]. Some original attempts to exploit the mathematical structure of the nonlinear pose graph optimization have been recently proposed in [20], [21], [30], [33]. In [30], Rizzini presented an interesting closed-form solution and showed relevant insights on the problem, under the hypothesis that the 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 (i) if the approach admits solution in general (Section III-B in [30]), (ii) what is the computational cost of the approach, (iii) what are the effects of a singular 4 by 4 description of the covariance matrices (due to *overparametrization*). In [20], Huang *et al.* discussed the convexity properties of SLAM and they drew conclusions about the importance of the orientation measurements, that are also confirmed by the results presented in this paper. More recently, Huang *et al.* investigated the number of minima in optimization-based SLAM problems [21], whereas in [33], Sünderhauf and Protzel studied solutions for rejecting outlying constraints in the pose graph.

Although the 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: the 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*). In this work we consider the batch SLAM problem in a planar setup and we provide two main contributions: a *linear approximation* for pose graph configuration and a technique for mitigating the *orientation wraparound problem*. The linear 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 in a global frame the relative position measurements. In the third phase, using the outcome of the first two phases, we solve a linear problem including robot poses. The proposed approximation is based on mild assumptions on the structure of the involved covariance matrices; it can be computed in closed-form and requires no initial guess. We show that the linear approximation is equivalent to a Gauss-Newton step around a suboptimal configuration estimate: such estimate is already close to the actual solution in practice, making the approach less prone to incur in local minima. Although implemented in Matlab, in most of the tests the computation of the linear approximation is faster than C++ implementations of state-of-the-art approaches. Moreover, experimental evidence demonstrates that the approximation is accurate in practice, then it can be used in place of the exact solution or for bootstrapping nonlinear techniques. As a by-product of our result we also show that (i) some steps of the computation of the linear approximation can be solved in incremental fashion, possibly performed online during robot operation; (ii) if convenient, some steps of the approach can be computed using expressions whose complexity depends on the number of loop closings in the pose graph.

The second contribution regards the orientation wraparound problem. We show that, in the proposed formulation, the wraparound problem has a clear meaning (incapability of discerning the number of 2π turns performed between two place-revisiting episodes); moreover, we discuss an approach for alleviating the issue. The structure of the cycles in the pose graph is shown to play a crucial role in the occurrence of the wraparound, and the theoretical development confirms empirical observations reported in related work [15].

We remark that the idea of a linear initialization is not new in literature. For instance, in computer vision, linear methods based on algebraic errors (e.g., [17], [29]) are often employed for bootstrapping nonlinear techniques (i.e., bundle adjustment [17]). Similar intuitions can be also found within the SLAM community [11]. The present paper extends our previous works [7] and [8]; in particular we improved the mentioned articles by including (i) a recursive procedure for orientation estimation, (ii) some insights on the computation of the *regularization* terms, (iii) a discussion on the orientation wraparound problem, (iv) an approach for computing the orientation estimates whose complexity depends on the number of loop-closings, and (v) several experimental results with an evaluation of accuracy and computational effort of the approach.

The rest of the article is organized as follows. Problem formulation is presented in Section II. The proposed linear approximation is presented in Section III; Section IV contains some original insights on the *orientation wraparound problem*. Experimental results are reported in Section V; conclusions are drawn in Section VI.

Notation and Preliminaries. $\mathbf{I}_n$ denotes the $n \times n$ identity matrix, $\mathbf{0}_n$ denotes a (column) vector of all zeros of dimension n. $M_{n\times m}$ denotes a matrix with n rows and m columns and $\otimes$ denotes the Kronecker product. The cardinality of a generic set S is written as $|S|$. 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 ordered 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*) [22]. The number of nodes and edges are denoted with $n+1$ (the reason for this choice will be clear later) and m, respectively, i.e., $|\mathcal{V}| = n+1$ and $|\mathcal{E}| = m$. The *incidence matrix* $\mathcal{A}$ of a directed graph is a matrix in $\mathbb{R}^{(n+1)\times m}$ 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 referred to as *chords* [9].

II. Problem Formulation

Let $\mathcal{V} = \{v_0, \ldots, v_n\}$ be a set of $n+1$ nodes (representing subsequent poses assumed by a mobile robot) and let $\mathcal{P} = \{p_0, \ldots, p_n\}$ denote a corresponding set of absolute poses in a planar setup, i.e., $p_i = [\rho_i^\top \ \theta_i]^\top \in$ SE(2), where $\rho_i = [x_i \ y_i]^\top \in \mathbb{R}^2$ is the Cartesian position of the i-th node, and θ_i is its orientation. We shall call $\mathcal{P}$ a *configuration* of nodes. Suppose that it is possible to measure the relative pose between some nodes' pairs, say nodes (i,j); in particular node i can measure the pose of j in its local reference frame $\mathcal{R}_i$:

$$\bar{\xi}_{ij} = p_j \ominus p_i,$$

where $\ominus$ is a standard *pose compounding operator* that can be rewritten in explicit form as:

$$p_j \ominus p_i \doteq \begin{bmatrix} R_i^\top(\rho_j - \rho_i) \\ \theta_j - \theta_i \end{bmatrix}, \tag{1}$$

being $R_i \in \mathbb{R}^{2\times 2}$ a planar rotation matrix of an angle θ_i. Since relative pose measurements are affected by noise, the measured quantities are in the form $\xi_{ij} = \bar{\xi}_{ij} + \epsilon_{ij}$[1], where $\epsilon_{ij} \in \mathbb{R}^3$ is a zero mean Gaussian noise, i.e., $\epsilon_{ij} \sim \mathcal{N}(\mathbf{0}_3, P_{ij})$, being P_{ij} a 3 by 3 covariance matrix. In practice, we can distinguish two kinds of relative pose measurements:

- *odometric constraints*: relative measurements between poses assumed by the robot at subsequent 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 has to 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).

In this context, we assume the constraints to be given, since the reliable determination of both odometric and loop closing constraints is still an active research topic [6], [32], whose implications are out of the scope of the present article.

From the knowledge of the relative pose measurements and the corresponding uncertainty, the robot is required to estimate the configuration $\mathcal{P}$ in a given reference frame $\mathcal{R}_0$. By convention, we set the initial pose of the robot to be the origin of such reference frame, i.e., $p_0 = [0 \ 0 \ 0]^\top$. In topological graph theory the problem is also referred to as graph *embedding*, *realization* or *drawing*, depending on the context and on problem constraints [16].

The problem can be naturally modeled using graph formalism: each node in the set $\mathcal{V}$ corresponds to a vertex of a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (often referred to as *pose graph*), where $\mathcal{E}$ (graph edges) is the set containing the unordered node pairs (i,j) such that a relative pose measurement exists between 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. We denote with Ξ the set of all available measurements, i.e., $\Xi = \{\xi_{ij}, (i,j) \in \mathcal{E}\}$. The cardinality of the set Ξ is $|\Xi| = |\mathcal{E}| = m$.

Therefore, our objective is to estimate a configuration $\mathcal{P}^* = \{p_1^*, \ldots, p_n^*\}$, that maximizes the likelihood of the observations. It is common in literature to assume independence among observations, hence we can write the log-likelihood function of a configuration as:

$$\ln \mathcal{L}(\mathcal{P}|\Xi) = \sum_{(i,j)\in\mathcal{E}} \ln \psi(\xi_{ij}|\mathcal{P}), \tag{2}$$

[1] A more formal definition of the uncertainty is $\epsilon_{ij} = \phi^{-1}(\xi_{ij} \ominus \bar{\xi}_{ij})$, where ϕ 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.

where $\psi(\xi_{ij}|\mathcal{P})$ is the conditional probability density of the measurement ξ_{ij} given nodes' configuration. Since we assumed that the involved densities are Gaussians, it is easy to demonstrate that maximizing the likelihood function (2) is equivalent to minimize the sum of the weighted residual errors $r_{ij}(\mathcal{P})$:

$$\begin{aligned} f(\mathcal{P}) = \textstyle\sum_{(i,j)\in\mathcal{E}} (p_j \ominus p_i - \xi_{ij})^\top P_{ij}^{-1}(p_j \ominus p_i - \xi_{ij}) = \\ = \textstyle\sum_{(i,j)\in\mathcal{E}} r_{ij}^\top(\mathcal{P})P_{ij}^{-1}r_{ij}(\mathcal{P}). \end{aligned} \tag{3}$$

Ordering the available measurements from 1 to m, we can define the *measurement vector* $\xi = [\xi_1^\top \; \xi_2^\top \; \ldots \; \xi_m^\top]^\top$ and the corresponding covariance matrix P, that is a block diagonal matrix in which the k-th 3 by 3 diagonal block contains the covariance matrix P_k of the k-th relative pose measurement. Similarly, we can build the *residual error vector* $r = [r_1^\top \; r_2^\top \; \ldots \; r_m^\top]^\top$, comprising the residual errors $r_{ij}(\mathcal{P})$ for each available measurement. According to the previous definitions we can write (3) in compact form as:

$$f(\mathcal{P}) = r^\top(\mathcal{P})P^{-1}r(\mathcal{P}). \tag{4}$$

The batch SLAM problem is hence formulated as a minimization of the nonlinear non-convex cost function (4), i.e., the optimal configuration is $\mathcal{P}^* = \arg\min f$. The nonlinearity of the function $f(\mathcal{P})$ 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 employ nonlinear optimization techniques for solving the problem at hand.

Iterative Nonlinear Optimization

In this section we briefly recall the basic idea behind nonlinear optimization techniques that are used in most of the state-of-the-art iterative approaches to graph-based SLAM. As mentioned before, such algorithms work iteratively by successively optimizing local convex approximations of the cost function (4). Let $\mathcal{P}^{(\tau)}$ be the configuration computed by the algorithm at iteration τ, being $\mathcal{P}^{(0)}$ the given initial guess. At each iteration, nonlinear techniques linearize $r(\mathcal{P})$ around the current solution $\mathcal{P}^{(\tau)}$, obtaining

$$r(\mathcal{P}) \simeq r(\mathcal{P}^{(\tau)}) + J_r\delta_p(\tau),$$

where $\delta_p(\tau) \in \mathbb{R}^{3n}$ is the displacement from the linearization point (it does not include the node v_0 which was assumed to be known), and $J_r = J_r(\mathcal{P}^{(\tau)}) \in \mathbb{R}^{3m\times 3n}$ is the Jacobian of the residual error function $r(\cdot)$. Therefore, related work approximates the cost function (4) around $\mathcal{P}^{(\tau)}$ with the convex objective:

$$f \simeq \left(r(\mathcal{P}^{(\tau)}) + J_r\delta_p(\tau)\right)^\top P^{-1}\left(r(\mathcal{P}^{(\tau)}) + J_r\delta_p(\tau)\right).$$

The previous cost function is quadratic in the unknown $\delta_p(\tau)$ and its minimizer corresponds to the solution of the *normal equation* [28]:

$$\left(J_r^\top P^{-1}J_r\right)\delta_p(\tau) = -J_r^\top P^{-1}r(\mathcal{P}^{(\tau)}). \tag{5}$$

Thus, at each iteration of the nonlinear least-squares method presented so far, the configuration is refined using the local correction $\delta_p(\tau)$ 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 (5). 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). The latter will be lately referred to as *indirect* techniques, since the term *iterative* already resembles the need of repeatedly approximating the original cost function with a suitable convex approximation.

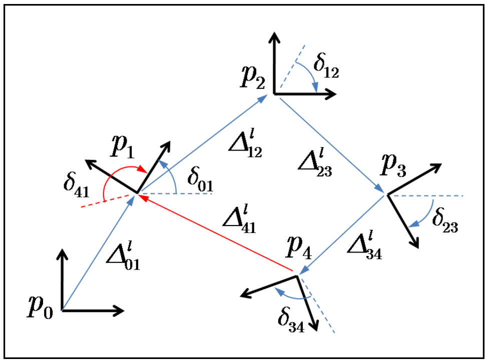


Fig. 1. Pose graph with 5 nodes: a reference frame is attached to each node to define the corresponding pose; odometric constraints are shown in blue, whereas an example of loop closing constraint is drawn in red.

III. A Linear Approximation for SLAM

We first observe that each relative pose measurement comprises a vector in $\mathbb{R}^2$ (*relative position*) corresponding to the first two components of ξ_{ij}, in (1), and a scalar (*relative orientation*), see also the toy example in Figure 1. Therefore we can rewrite each measurement as $\xi_{ij} = [(\Delta^l_{ij})^\top \; \delta_{ij}]^\top$, where the superscript l denotes that the relative position vector is expressed in a local frame. According to the notation introduced so far the cost function (3) can be rewritten as:

$$f(\mathcal{P}) = \textstyle\sum_{(i,j)\in\mathcal{E}} \begin{bmatrix} R_i^\top(\rho_j-\rho_i)-\Delta^l_{ij} \\ (\theta_j-\theta_i)-\delta_{ij} \end{bmatrix}^\top P_{ij}^{-1} \begin{bmatrix} R_i^\top(\rho_j-\rho_i)-\Delta^l_{ij} \\ (\theta_j-\theta_i)-\delta_{ij} \end{bmatrix}.$$

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

$$\begin{aligned} f(\mathcal{P}) = {} & \textstyle\sum_{(i,j)\in\mathcal{E}} \left[R_i^\top(\rho_j-\rho_i)-\Delta^l_{ij}\right]^\top P^{-1}_{\Delta^l_{ij}} \left[R_i^\top(\rho_j-\rho_i)-\Delta^l_{ij}\right] + \\ & + \textstyle\sum_{(i,j)\in\mathcal{E}} \left[(\theta_j-\theta_i)-\delta_{ij}\right]^\top P^{-1}_{\delta_{ij}} \left[(\theta_j-\theta_i)-\delta_{ij}\right]. \end{aligned} \tag{6}$$

In order to put the previous formulation in a more compact form, let us number the nodes pairs, for which a relative pose measurement is available, from 1 to m; let us stack all the relative position measurements in the vector $\Delta^l = [(\Delta^l_1)^\top \; (\Delta^l_2)^\top \; \ldots \; (\Delta^l_m)^\top]^\top$, and all the relative orientation measurements in the vector $\delta = [\delta_1 \; \delta_2 \; \ldots \; \delta_m]^\top$. Repeating the same procedure for all the positions and orientations assumed by the robot until the current time we get the *nodes' position* $\rho^\dagger = [\rho_0^\top \; \rho_1^\top \; \ldots \; \rho_n^\top]^\top$ and the *nodes' orientation* $\theta^\dagger = [\theta_0 \; \theta_1 \; \ldots \; \theta_n]^\top$. Therefore it can be verified that equation (6) can be rewritten as:

$$\begin{aligned} f(\mathcal{P}) = {} & \left(\mathcal{A}_2^\top \rho^\dagger - R\Delta^l\right)^\top \left(R P_{\Delta^l} R^\top\right)^{-1} \left(\mathcal{A}_2^\top \rho^\dagger - R\Delta^l\right) + \\ & + \left(\mathcal{A}^\top \theta^\dagger - \delta\right)^\top P_\delta^{-1} \left(\mathcal{A}^\top \theta^\dagger - \delta\right), \end{aligned} \tag{7}$$

where $\mathcal{A}$ is the incidence matrix of the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, $\mathcal{A}_2 = \mathcal{A} \otimes \mathbf{I}_2$ is an expanded form of the incidence matrix [2], $P_{\Delta^l} = \text{diag}(P_{\Delta^l_1}, P_{\Delta^l_2}, \ldots, P_{\Delta^l_m})$, $P_\delta = \text{diag}(P_{\delta_1}, P_{\delta_2}, \ldots, P_{\delta_m})$, and $R = R(\theta)$ 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 R are in positions $(2k-1, 2k-1), (2k-1, 2k), (2k, 2k-1), (2k, 2k), \; k = 1, \ldots, m$, and the k-th diagonal block contains the rotation matrix converting the k-th relative position measurement in the global frame. The expanded matrix $\mathcal{A}_2$, or the more general form $\mathcal{A}_u = \mathcal{A} \otimes \mathbf{I}_u$, allows to cope with the case in which the node variables are vector-valued quantities in $\mathbb{R}^u$: the overall structure of the matrix is the same as $\mathcal{A}$, but the zero elements in $\mathcal{A}$ are replaced with a null matrix $\mathbf{0}_{u\times u}$, whereas the elements of value $+1$ and -1 are replaced by $\mathbf{I}_u$ and $-\mathbf{I}_u$, respectively.

Equation (7) can be further simplified by recalling the choice of the reference frame $\rho_0 = [0 \;\; 0]^\top$ and $\theta_0 = 0$, leading to the following relation:

$$\begin{aligned} f(\mathcal{P}) = \left(A_2^\top \rho - R\Delta^l\right)^\top \left(R P_{\Delta^l} R^\top\right)^{-1} \left(A_2^\top \rho - R\Delta^l\right) + \\ + \left(A^\top \theta - \delta\right)^\top P_\delta^{-1} \left(A^\top \theta - \delta\right), \end{aligned} \tag{8}$$

where ρ and θ are the vectors obtained from $\rho^\dagger$ and $\theta^\dagger$ by deleting the elements ρ_0 and θ_0, respectively, whereas A and A_2 are *reduced* incidence matrices, obtained by deleting the rows corresponding to θ_0 and ρ_0 from $\mathcal{A}$ and $\mathcal{A}_2$, respectively.

We observe that the minimization of the cost function (8) is, by definition, equivalent to find the solution (in the least-squares sense) to the following system of equations:

$$\left\{ \begin{array}{c} A_2^\top \rho = R(\theta)\Delta^l \\ A^\top \theta = \delta \end{array} \right. . \tag{9}$$

Roughly speaking, since system (9) 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 (8) attains the minimum of the weighted residual errors. The nonlinear nature of the problem is connected with the matrix $R(\theta)$, which contains trigonometric functions of nodes' orientation.

Before presenting our closed-form approximation, we underline that, when the orientation of the robot is known, the estimation problem becomes linear. Under this assumption, the second equation in problem (9) disappears and the pose estimation reduces to *positioning* (only ρ has to be estimated). If θ is equal to a known θ^*, we can rewrite (9) as:

$$A_2^\top \rho = R^* \Delta^l,$$

with $R^* = R(\theta^*)$. The previous is a linear estimation problem and it can be uniquely solved if at least the position of one node is known and the graph is weakly connected [1], [2]. In case no node position is known, the problem will have infinite solutions, corresponding to arbitrary roto-translations of the configuration. If we define $\Delta^g = R^* \Delta^l$ and $P_{\Delta^g} = (R^*) P_{\Delta^l} (R^*)^\top$, the *Best Linear Unbiased Estimator* (BLUE) for the unknown node positions ρ is [27]:

$$\hat{\rho}^* = (A_2 P_{\Delta^g}^{-1} A_2^\top)^{-1} A_2 P_{\Delta^g}^{-1} \Delta^g,$$

and the corresponding covariance matrix is $(A_2 P_{\Delta^g}^{-1} A_2^\top)^{-1}$. We notice that Δ^g and P_{Δ^g} are respectively the relative position measurements and the measurement covariance, expressed in the global frame, hence the linearity of this simplified problem stems from the capability of expressing the relative position measurements in a common frame. Such setup has been extensively investigated in both centralized and distributed contexts, with application to multi agent localization [1]; 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, which contains the first key contribution of the paper, 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.

Lemma 1 (Properties of 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 T is the spanning tree comprising all odometric constraints, the edges corresponding to loop closing constraints are chords of $\mathcal{G}$, with respect to T.

Proof. The proof of the first claim is trivial since the path connecting all the nodes is actually the trajectory traveled by the robot; in this sense the edges corresponding to the odometric constraints assure connectivity. Moreover, by definition of odometric constraints, a graph with $n+1$ nodes has n odometric constraints, and, since they connect all nodes, they form a spanning tree, say T; accordingly, the remaining $m-n$ constraints correspond to chords in $\mathcal{G}$ with respect to T. □

A. Linear Estimate for SLAM

We start by presenting the procedure for computing the proposed linear approximation of pose graph configuration. The procedure will be discussed in detail in the following sections, that also provide an assessment of the proposed linear approximation.

Procedure *1 (Linear Approximation of Pose Graph Configuration):* A linear approximation of pose graph configuration can be obtained by applying the following algorithm, structured in three phases:

1) Solve the following linear estimation problem:

$$A^\top \theta = \delta, \tag{10}$$

from which the suboptimal orientation estimate $\hat{\theta}$ and its covariance matrix $P_{\hat{\theta}}$ can be obtained.

2) Estimate the relative position measurements in the global reference frame:

$$z = \begin{bmatrix} \hat{R}\Delta^l \\ \hat{\theta} \end{bmatrix} = \begin{bmatrix} g_1(\Delta^l, \theta) \\ g_2(\theta) \end{bmatrix}_{\theta=\hat{\theta}}, \tag{11}$$

with $\hat{R} = R(\hat{\theta})$; compute the corresponding uncertainty (preserving the correlation with the orientation estimate):

$$P_z = H \begin{bmatrix} P_{\Delta^l} & \mathbf{0}_{2m\times n} \\ \mathbf{0}_{2m\times n}^\top & P_{\hat{\theta}} \end{bmatrix} H^\top, \tag{12}$$

where H is the Jacobian of the transformation in (11), i.e.,:

$$H \doteq \begin{bmatrix} \frac{\partial g_1}{\partial \Delta^l} & \frac{\partial g_1}{\partial \theta} \\ \frac{\partial g_2}{\partial \Delta^l} & \frac{\partial g_2}{\partial \theta} \end{bmatrix} = \begin{bmatrix} \hat{R} & J \\ \mathbf{0}_{2m\times n}^\top & \mathbf{I}_n \end{bmatrix}. \tag{13}$$

3) Solve the following linear estimation problem in the unknown $p = [\rho^\top \ \theta^\top]^\top$, given z, in (11), and P_z, in (12):

$$z = \begin{bmatrix} A_2^\top & \mathbf{0}_{2m\times n} \\ \mathbf{0}_{n\times 2n} & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \rho \\ \theta \end{bmatrix} = B^\top p, \tag{14}$$

from which the linear approximation of the pose graph configuration $p^* = [(\rho^*)^\top \ \ (\theta^*)^\top]^\top$ can be retrieved.

□

For sake of readability, the phases of Procedure 1 are discussed in detail in Sections III-B, III-C, III-D, that also provide an assessment of the resulting linear approximation.

B. Phase 1: Linear Orientation Estimation

Observing the structure of problem (9) it can be seen that the second part of the system, including the relative orientation measurements, is linear in the unknown variable θ. The first phase of Procedure 1 requires the solution of such subset of constraints; this is a standard linear estimation problem: given the matrix A, the measurements δ and the corresponding covariance matrix P_δ, the objective is to provide an estimate $\hat{\theta}$ of the unknown θ. According to well known results in linear estimation theory [27], the Best Linear Unbiased Estimator (BLUE) for θ in (10) is:

$$\hat{\theta} = (AP_{\delta}^{-1}A^{\top})^{-1}AP_{\delta}^{-1}\delta, \tag{15}$$

and the corresponding covariance is:

$$P_{\hat{\theta}} = (AP_{\delta}^{-1}A^{\top})^{-1}. \tag{16}$$

The following result establishes existence and uniqueness of solution of problem (10). We recall that similar results can be found in literature, with different proofs, see [2] and the references therein.

Proposition 1 (Uniqueness of the solution for phase 1): The solution of the first phase of Procedure 1 is unique if $\mathcal{G}$ is connected and a node orientation is supposed to be known.

Proof. The incidence matrix $\mathcal{A}$ of a connected graph $\mathcal{G}$ has rank n (with $|\mathcal{V}| = n+1$) and becomes full rank as soon as one row is deleted [9]. As we have just shown in (8), assuming a node to have known orientation (e.g., the first node is assumed to be the absolute reference frame) allows to reduce the incidence matrix by deleting the corresponding row; therefore, the reduced incidence matrix A is full rank. The information matrix P_{δ}^{-1} is positive-definite by definition. Then $AP_{\delta}^{-1}A^{\top}$ is positive-definite, hence invertible [18]. If the matrix is invertible, the solution of the linear problem can be uniquely determined as $\hat{\theta} = (AP_{\delta}^{-1}A^{\top})^{-1}AP_{\delta}^{-1}\delta$. □

We already observed that, once the absolute orientation of the robot is known, also the first equation in (9) becomes linear in the unknown ρ. Therefore, using $\hat{\theta}$ as the true nodes' orientation, we could also compute an estimate for nodes' position $\hat{\rho}$, using linear estimation framework:

$$\hat{\rho} = \left[A_2\left(\hat{R}P_{\Delta^l}\hat{R}^{\top}\right)^{-1}A_2^{\top}\right]^{-1}A_2\left(\hat{R}P_{\Delta^l}\hat{R}^{\top}\right)^{-1}\hat{R}\Delta^l, \tag{17}$$

where $\hat{R} = R(\hat{\theta})$. Also in this case, the proof of existence and uniqueness of the solution directly stems from Proposition 1. However, the first equation in (9) also constraints the orientations of the robot, thus the estimate $\hat{p} = [\hat{\rho}^{\top}\ \hat{\theta}^{\top}]^{\top}$ 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 (8) with respect to θ, to substitute this suboptimal solution in the first summand, and to optimize it with respect to the position variables ρ. The obtained solution, however, does not correspond to a minimum of the overall cost function.

As a further contribution, we report in Appendix C an approach for computing $\hat{\theta}$ and $\hat{\rho}$, whose complexity depends on the number of loop closings in the pose graph.

C. Phase 2: First-order Error Propagation

According to linear estimation theory, when solving problem (10) we can compute the actual covariance of the estimate (16), that, together with the mean value (15), describes the posterior of θ given the orientation measurements; moreover, as input of the problem we have Δ^l and the corresponding covariance. The second phase of Procedure 1 uses this information to compute an estimate of the relative position measurements in the global reference frame: the vector z is in the form $z = [(\Delta^g)^{\top}\ \hat{\theta}^{\top}]^{\top}$ where $\Delta^g = \hat{R}\Delta^l$ is the vector containing the relative nodes' positions expressed in the absolute reference frame $\mathcal{R}_0$. z is the output of a nonlinear transformation involving $\hat{\theta}$ and Δ^l. The corresponding covariance matrix can be obtained by a first-order propagation of the uncertainty; hence we can rewrite (12) in explicit form as:

$$P_z = \begin{bmatrix} P_{\Delta^g} + JP_{\hat{\theta}}J^{\top} & JP_{\hat{\theta}} \\ P_{\hat{\theta}}J^{\top} & P_{\hat{\theta}} \end{bmatrix},$$

where $P_{\Delta^g} \doteq \hat{R}P_{\Delta^l}\hat{R}^{\top}$, whereas J and $P_{\hat{\theta}}$ are defined as in (13) and (16). The following proposition will be useful later for assuring the uniqueness of the outcome of the proposed approach.

Proposition 2 (Positive-definiteness of P_z): The covariance matrix of the data vector z, P_z, computed in the second phase of Procedure 1, is positive-definite.

Proof. The covariance matrix $P_{\hat{\theta}}$ was demonstrated to be positive-definite in Proposition 1; the matrix P_{Δ^l} is positive-definite by definition. Therefore the matrix diag$(P_{\Delta^l}, P_{\hat{\theta}})$ is also positive-definite. As in Proposition 1, we now want to demonstrate that the matrix H in (12) is full rank. Because of the block structure of H, it follows that $\det(H) = \det(\hat{R})\det(\mathbf{I}_n) = \det(\hat{R})$ [18]. $\hat{R}$ is a block diagonal matrix, whose 2 by 2 blocks are rotation matrix, hence $\det(\hat{R}) = \prod_{(i,j)\in\mathcal{E}} \det(R_i) = 1$. Therefore $\det(H) = 1$ and H is full rank. Accordingly, the matrix $H \cdot$ diag$(P_{\Delta^l}, P_{\hat{\theta}}) \cdot H^\top$ is positive-definite [18]. □

The previous result assures that there is no rank loss in the covariance propagation. The trick of including the orientation estimates in z is useful for preserving 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 Procedure 1.

D. Phase 3: Linear Pose Estimation

We now want to show that the last phase allows to correct the suboptimal configuration estimate $\hat{p} \doteq [\hat{\rho}^\top \;\; \hat{\theta}^\top]^\top$, leading it towards a minimum of the cost function, i.e.

$$\theta^* = \hat{\theta} + \tilde{\theta}, \qquad \rho^* = \hat{\rho} + \tilde{\rho},$$

in which $\hat{\theta}$ and $\hat{\rho}$ are computed as in equations (15) and (17), while $\tilde{\theta}$ and $\tilde{\rho}$ are local correction terms. In order to proceed in the demonstration we need to compute $p^* = [(\rho^*)^\top \;\; (\theta^*)^\top]^\top$ as solution of the linear system (14):

$$p^* = \begin{bmatrix} \rho^* \\ \theta^* \end{bmatrix} = (BP_z^{-1}B^\top)^{-1}BP_z^{-1}z. \tag{18}$$

First of all, let us state the following result on existence and uniqueness of the solution of the linear estimation problem in phase 3.

Proposition 3 (Uniqueness of the solution for phase 3): The solution of the third phase of Procedure 1 is unique if $\mathcal{G}$ is connected and a node pose is supposed to be known.

Proof. From Proposition 2, the matrix P_z is positive-definite, and so is its inverse P_z^{-1}. Because of the structure of matrix B, it holds rank(B) = rank(A_2) + rank$(\mathbf{I}_n)$ = rank$(A_2) + n$. We have already reported in Proposition 1 that the rank of the reduced incidence matrix A is n. As a consequence, rank(A_2) = rank$(A \otimes \mathbf{I}_2)$ = rank(A)rank$(\mathbf{I}_2) = 2n$, It follows that the rank of B is $3n$, therefore, if P_z^{-1} is positive-definite and B is full rank, then $BP_z^{-1}B^\top$ is positive-definite, hence invertible [18]. Therefore, the solution of the linear problem can be uniquely determined. □

Now we can write in explicit form ρ^* and θ^*; for this purpose let us compute the matrix P_z^{-1}:

$$P_z^{-1} = \begin{bmatrix} P_{\Delta^g}^{-1} & -P_{\Delta^g}^{-1}J \\ -J^\top P_{\Delta^g}^{-1} & P_{\hat{\theta}}^{-1} + J^\top P_{\Delta^g}^{-1}J \end{bmatrix}.$$

The previous inverse can be performed using blockwise matrix inversion [18]. We can then compute the matrix $\Omega_{p^*} \doteq (BP_z^{-1}B^\top)$:

$$\Omega_{p^*} = \begin{bmatrix} P_{\hat{\rho}}^{-1} & -A_2P_{\Delta^g}^{-1}J \\ -J^\top P_{\Delta^g}^{-1}A_2^\top & P_{\hat{\theta}}^{-1} + J^\top P_{\Delta^g}^{-1}J \end{bmatrix},$$

where $P_{\hat{\rho}} \doteq (A_2P_{\Delta^g}^{-1}A_2^\top)^{-1}$. The inverse of Ω_{p^*}, namely P_{p^*}, is in the form:

$$P_{p^*} = \begin{bmatrix} P_{\rho^*} & P_{\rho^*,\theta^*} \\ P_{\rho^*,\theta^*}^\top & P_{\theta^*} \end{bmatrix},$$

where:

$$\begin{aligned} &P_{\theta^*} \doteq \left(P_{\hat{\theta}}^{-1} + J^\top P_{\Delta^g}^{-1} J - J^\top P_{\Delta^g}^{-1} A_2^\top P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} J\right)^{-1} \\ &P_{\rho^*} = P_{\hat{\rho}} + P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} J P_{\theta^*} J^\top P_{\Delta^g}^{-1} A_2^\top P_{\hat{\rho}} \\ &P_{\rho^*,\theta^*} = P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} J P_{\theta^*}. \end{aligned} \tag{19}$$

From the expression of P_{p^*} and P_z^{-1} it is straightforward to compute the estimates (18) as:

$$\begin{gathered} \theta^* = (A P_\delta^{-1} A^\top)^{-1} A P_\delta^{-1} \delta + \\ + P_{\theta^*} J^\top P_{\Delta^g}^{-1} \left(A_2^\top P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} - \mathbf{I}_{2m}\right) \hat{R} \Delta^l, \end{gathered} \tag{20}$$

and:

$$\begin{gathered} \rho^* = \left[A_2 \left(\hat{R} P_{\Delta^l} \hat{R}^\top\right)^{-1} A_2^\top\right]^{-1} A_2 \left(\hat{R} P_{\Delta^l} \hat{R}^\top\right)^{-1} \hat{R} \Delta^l + \\ + P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} J P_{\theta^*} J^\top P_{\Delta^g}^{-1} \left(A_2^\top P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} - \mathbf{I}_{2m}\right) \hat{R} \Delta^l. \end{gathered}$$

Notice that the previous equation provides a closed-form expression of the outcome of Procedure 1. In Section V-A we will point out that there is a more efficient way to compute the proposed approximation, although the explicit solution (20) is useful for the following development. By simple inspection it is possible to verify that the obtained estimate θ^* is already in the form $\theta^* = \hat{\theta} + \tilde{\theta}$, since the first summand in (20) coincides with (15). The same consideration holds for ρ^* with respect to equation (17). Therefore we can write $\tilde{\theta}$ and $\tilde{\rho}$ as:

$$\begin{aligned} &\tilde{\theta} = P_{\theta^*} J^\top P_{\Delta^g}^{-1} \left(A_2^\top P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} - \mathbf{I}_{2m}\right) \hat{R} \Delta^l \\ &\tilde{\rho} = P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} J P_{\theta^*} J^\top P_{\Delta^g}^{-1} \left(A_2^\top P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} - \mathbf{I}_{2m}\right) \hat{R} \Delta^l. \end{aligned} \tag{21}$$

We are now ready to state the following key result.

Theorem 1 (Linear Approximation): Procedure 1 is equivalent to computing an initial guess of network configuration and performing a local optimization step, providing an approximate solution to the nonlinear optimization problem (8). In particular, the third phase produces a local correction of a suboptimal configuration estimate computed in the first phase, leading it towards a minimum of the cost function (8).

Proof. We have already shown that the final solution is composed by the suboptimal solution plus a correction term. Now the demonstration reduces to verify that $\tilde{p} = [\tilde{\rho}^\top \ \tilde{\theta}^\top]^\top$ is a local solution of our optimization problem around $\hat{p} = [\hat{\rho}^\top \ \hat{\theta}^\top]^\top$. In order to prove this point we compute a first-order approximation of the residual errors in (8) around the suboptimal solution $\hat{p}$:

$$\begin{gathered} f(\mathcal{P}) \approx \left(A_2^\top \hat{\rho} + A_2^\top \delta_\rho - \hat{R} \Delta^l - J \delta_\theta\right)^\top \\ \left(\hat{R} P_{\Delta^l} \hat{R}^\top\right)^{-1} \left(A_2^\top \hat{\rho} + A_2^\top \delta_\rho - \hat{R} \Delta^l - J \delta_\theta\right) + \\ + \left(A^\top \hat{\theta} + A^\top \delta_\theta - \delta\right)^\top P_\delta^{-1} \left(A^\top \hat{\theta} + A^\top \delta_\theta - \delta\right), \end{gathered} \tag{22}$$

where δ_θ and δ_ρ are the displacements from the linearization point and the covariance matrix $R P_{\Delta^l} R^\top$ is evaluated in $\hat{\theta}$. This convex approximation can now be minimized by taking the first derivative with respect to the optimization variables δ_θ and δ_ρ and imposing it to be zero. The minimum of (22) corresponds to (a complete derivation is reported in Appendix A):

$$\begin{aligned} &\delta_\theta = P_{\theta^*} J^\top P_{\Delta^g}^{-1} \left(A_2^\top P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} - \mathbf{I}_{2m}\right) \hat{R} \Delta^l \\ &\delta_\rho = P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} J P_{\theta^*} J^\top P_{\Delta^g}^{-1} \left(A_2^\top P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} - \mathbf{I}_{2m}\right) \hat{R} \Delta^l, \end{aligned} \tag{23}$$

which can be seen to coincide, respectively, with $\tilde{\theta}$ and $\tilde{\rho}$, see (21), thus proving our thesis. □

We remark that $\hat{\rho}$ is not actually computed by the proposed approach, and it was only derived in this section for the purpose of the demonstration. Moreover, in practice, the suboptimal orientation estimate $\hat{\theta}$ is quite accurate, therefore the third phase is expected to lead the approximation towards a global optimum of problem (8). In the experimental section, we will show that the proposed approach allows to attain the same accuracy of an iterative method in which the local optimization step is repeated several times to correct an inaccurate initial guess (e.g., odometry).

E. Regularization and Existence of the Correction Factors

We now discuss a crucial point of the proposed approach, which is connected with 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 2π. 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 2π actually correspond to the same orientation, hence tries to impose contrasting constraints, producing meaningless results. An easy solution to the previous problem consists in adding a correction factor, in the form $2k\pi,\ k \in \mathbb{Z}$, to some orientation measurements. These correction factors do not alter measurement content, because of the periodicity of the angular information, but let the relative orientation measurements sum up to zero (this property will be lately referred to as *zero-sum* property). Hence the input data provided to the problem solver are consistent and the estimated configuration is correct.

In the rest of this section we prove the existence of suitable correction factors for any connected graph and we describe a methodology for retrieving such correction terms. Before presenting the main result (Theorem 3) let us introduce some specific concepts from graph theory [22].

A *cycle* is a subgraph in which every node appears in a 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 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 c_i is arbitrary).

Definition 1: Given a directed graph $\mathcal{G}$ and a spanning tree T of $\mathcal{G}$, a *fundamental circuit* is a circuit composed by a chord (i, j) of $\mathcal{G}$ with respect to T and the unique path in T connecting i and j. □

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 2: The set of the fundamental circuits of a directed graph constitutes a cycle basis for $\mathcal{G}$. □

The proof of the previous theorem can be found in several books of graph theory, see [9]. We already mentioned that a spanning tree 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 1: The dimension of the cycle space of a connected graph G is $\ell = m - n$, and it is usually called *cyclomatic* number of the graph [22]. □

Corollary 2: 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 T, and the last $m - n$ elements are the chords of $\mathcal{G}$ with respect T, the matrix containing all the vectors c_i corresponding to the fundamental circuits can be written as:

$$\mathcal{C} = [c_1\ c_2\ \ldots\ c_\ell]^\top = [\mathcal{B} \quad I_\ell]\,, \tag{24}$$

where I_ℓ is the identity matrix of dimension ℓ and $\mathcal{B}$ is a matrix with elements in $\{-1, 0, 1\}$. $\mathcal{C}$ 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 [9]. With slight abuse of notation, in the following, $\mathcal{C}$ will denote both the cycle basis and the 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 3 (Existence of correction factors): Given relative orientation measurements $\bar{\delta} = [\bar{\delta}_1 \; \bar{\delta}_2 \; \ldots \; \bar{\delta}_m]^\top$, such that the first n elements correspond to the edges of a given spanning tree T, and the last $m - n$ elements correspond to the chords of $\mathcal{G}$ with respect T, there exists a correction vector $\nu = [\nu_1 \; \nu_2 \; \ldots \; \nu_m]^\top$ so that the corrected measurements $\delta = [\bar{\delta}_1 - \nu_1 \;\; \bar{\delta}_2 - \nu_2 \; \ldots \; \bar{\delta}_m - \nu_m]^\top$ 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 [22]. Let us consider the cycle basis composed by the fundamental circuits; for the zero-sum property to hold true, the corrected measurements δ have to satisfy:

$$c_i^\top \delta = 0 \qquad \forall c_i \in \mathcal{C}. \tag{25}$$

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 (25) can be written in compact form using the cycle basis matrix:

$$\mathcal{C}\delta = \mathbf{0}_\ell. \tag{26}$$

According to the definition of regularized measurements we can rewrite (26) as:

$$\mathcal{C}(\bar{\delta} - \nu) = \mathbf{0}_\ell \Longrightarrow \mathcal{C}\nu = \mathcal{C}\bar{\delta}. \tag{27}$$

The right hand side will contain the sum of the original measurements for each fundamental circuit. In a noiseless case, the vector $\mathcal{C}\bar{\delta}$ contains terms in the form $2k\pi, \; k \in \mathbb{Z}$. Since the cycle basis matrix $\mathcal{C}$ can be computed from the graph and $\bar{\delta}$ is a given of the problem, the only unknown of (27) is ν and the existence of suitable correction terms is reduced the demonstration of existence of a solution to system (27). Recalling (24), it is easy to show that a solution to system (27), is $\nu = [\mathbf{0}_n^\top \;\; (\mathcal{C}\bar{\delta})^\top]^\top$:

$$\mathcal{C}\nu = [\mathcal{B} \;\; I_\ell]\,[\mathbf{0}_n^\top \;\; (\mathcal{C}\bar{\delta})^\top]^\top = \mathcal{C}\bar{\delta},$$

hence proving our thesis. □

The elements of δ are referred to as *regularized relative orientation measurements*. The process of compensating the relative orientation measurements is named *regularization*. 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 ν are zeros). Recalling Lemma 1, if we consider the spanning tree comprising the odometric constraints, we can then regularize the relative orientation measurements by simply correcting loop closing constraints; in particular, a loop closing relation, constraining two robot poses, has to be corrected taking into account the number of complete turns (2π 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. An explanatory example of regularization is reported in Section V-A. We remark that the correction terms are in the form $2k\pi, \; k \in \mathbb{Z}$, hence the regulation procedure does not alter the information content of the orientation measurement. It is now evident that, in the case of noisy relative measurements, condition

(25) cannot be met exactly: Procedure 1, in fact, will be in charge of compensating the measurement errors by minimizing a suitable cost function. Accordingly, the term $\mathcal{C}\bar{\delta}$ in (27) will not contain exact multiples of 2π. However, a simple rounding to the closest multiple of 2π allows to retrieve the desired correction factors. One may argue that, if the noise is large, it is not possible to discern the desired correction factors, since the rounding cannot compensate measurement errors; however this issue was not found to be relevant in common applications (see experimental section): as it will be clear in a while, the impossibility to determine the correct multiple of 2π 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 2π turns) since the previous visit. Note that this result also sheds some light on the so called *orientation wraparound* problem [15], which is known to prevent convergence in iterative approaches.

In order to reduce the influence of noise in the selection of the correction terms for regularization, one may use a *minimal cycle basis* [22], in place of the fundamental circuit basis. In the following section, instead, we consider a less obvious approach to alleviate the influence of noise on regularization. We lately refer to the regularization presented in this section as *static* regularization, whereas the approach of Section IV-B will be called *dynamic regularization*. *Dynamic regularization* is reported here for the sake of completeness, although it is often unnecessary in common SLAM problems.

IV. Alleviating the Orientation Wraparound Problem

As a conclusion of the theoretical contribution of the paper we now discuss how to reduce the risk of incurring in the orientation wraparound problem which constitutes a limitation of the state-of-the-art approaches to graph-based SLAM. In iterative solutions to graph-based SLAM, the orientation wraparound problem occurs when the error in the relative orientation of some nodes in the initial guess is bigger than π [15]. For the proposed approach, instead, it affects the capability of determining proper correction factors for regularization, as described in Section III-E. In both cases it depends on the amount of noise on the relative orientation measurements. Intuitive solutions to such problem consist in reducing orientation measurement noise or in limiting the dimension of the loops in the graph. These expedients, however, cannot be applied in general, since measurement accuracy may not be increased arbitrarily (at least without a corresponding surplus in price or complexity), and strict constraints on graph structure are not desirable, since we are envisioning to solve large-scale problems.

Instead, we now discuss how to prevent the wraparound issue without imposing further constraints on the problem at hand. In the following section we first present an approach for solving the first phase of Procedure 1 in a recursive way; then we discuss how this technique can reduce the risk of orientation wraparound.

A. Recursive Procedure for Linear Orientation Estimation

Let us consider a pose graph $\mathcal{G}[0]$; let us call $A[0]$, $\delta[0]$, and $P_\delta[0]$ the reduced incidence matrix, the regularized relative orientation measurements and the corresponding covariance matrix. Assume that we computed the orientation estimate $\hat{\theta}[0]$ and the corresponding covariance matrix $P_{\hat{\theta}}[0]$ according to (15) and (16). Now, assume that an edge is added to the graph. We shall call the augmented graph $\mathcal{G}[1]$, and the objective of this section is to discuss how to retrieve $\hat{\theta}[1]$ and $P_{\hat{\theta}}[1]$ (orientation estimate and corresponding covariance in $\mathcal{G}[1]$), exploiting the knowledge of $\hat{\theta}[0]$ and $P_{\hat{\theta}}[0]$.

We call $A[1]$, $\delta[1]$, and $P_\delta[1]$, the reduced incidence matrix, the relative orientation measurements and the corresponding covariance matrix, of graph $\mathcal{G}[1]$. Since we obtained the graph $\mathcal{G}[1]$ by adding a single edge to $\mathcal{G}[0]$, it is easy to verify the following relations:

$$A[1] = \left[\begin{array}{cc} A[0] & a \end{array}\right], \qquad \delta[1] = \left[\begin{array}{c} \delta[0] \\ \delta_{m+1} \end{array}\right], \qquad P_\delta[1] = \left[\begin{array}{cc} P_\delta[0] & \mathbf{0}_m \\ \mathbf{0}_m^\top & P_{\delta_{m+1}} \end{array}\right], \tag{28}$$

where a is a column vector with entries in $\{-1, 0, +1\}$, describing the edge added to $\mathcal{G}[0]$, whereas δ_{m+1} is the relative orientation measurement associated with the edge we added ($P_{\delta_{m+1}}$ is the corresponding variance). Equation (15) prescribes to compute the orientation estimate $\hat{\theta}[1]$ as follows:

$$\hat{\theta}[1] = \left(A[1]\; P_{\delta}^{-1}[1]\; A^{\top}[1]\right)^{-1}\; A[1]\; P_{\delta}^{-1}[1]\; \delta[1], \tag{29}$$

The most expensive computation for retrieving the solution (29) is the inversion of the information matrix $P_{\hat{\theta}}[1] \doteq \left(A[1]\;\; P_{\delta}^{-1}[1]\;\; A^{\top}[1]\right)$. We now want to show that this inverse needs not be computed from scratch but can be computed exploiting the knowledge of $P_{\hat{\theta}}[0]$. For this purpose, let us substitute the relations in (28) as follows:

$$\begin{aligned} P_{\hat{\theta}}[1] = \left(A[1]\; P_{\delta}^{-1}[1]\; A^{\top}[1]\right)^{-1} &= \left(\begin{bmatrix} A[0] & a \end{bmatrix} \begin{bmatrix} P_{\delta}[0] & \mathbf{0}_m \\ \mathbf{0}_m^{\top} & P_{\delta_{m+1}} \end{bmatrix}^{-1} \begin{bmatrix} A[0] & a \end{bmatrix}^{\top} \right)^{-1} = \\ &= \left(A[0]\; P_{\delta}^{-1}[0]\; A^{\top}[0] \;+\; a\; P_{\delta_{m+1}}^{-1}\; a^{\top}\right)^{-1} \end{aligned} \tag{30}$$

We now observe that $A[0]P_{\delta}^{-1}[0]A^{\top}[0]$ is, by definition, $P_{\hat{\theta}}[0]$. Then, applying the *matrix inversion lemma* [4], we simplify equation (30) as follows:

$$P_{\hat{\theta}}[1] = P_{\hat{\theta}}[0] - P_{\hat{\theta}}[0]\; a\; \left(P_{\delta_{m+1}} + a^{\top} P_{\hat{\theta}}[0] a\right)^{-1}\; a^{\top}\; P_{\hat{\theta}}[0], \tag{31}$$

where $\left(P_{\delta_{m+1}} + a^{\top} P_{\hat{\theta}}[0] a\right)$ is a scalar. From expression (31) we can then compute $\hat{\theta}[1]$ as:

$$\hat{\theta}[1] = P_{\hat{\theta}}[1]\; A[1]\; P_{\delta}^{-1}[1]\; \delta[1]. \tag{32}$$

Therefore, given the linear orientation estimate over a graph $\mathcal{G}[0]$, we can incrementally add edges and evaluate the suboptimal orientation estimate over the augmented graph $\mathcal{G}[1]$. In Appendix B we generalize the recursive estimation to the case in which also nodes are added to the graph: this can be of interest when the orientation estimate has to be computed online, i.e., during robot motion. In the following section, instead, we discuss how this recursive scheme can alleviate the orientation wraparound problem, within the proposed approach.

B. Orientation Estimate and Wraparound Problem

In this section we propose a second technique for regularizing the relative orientation measurements of a given pose graph $\mathcal{G}$. As in previous sections, we adopt the following notation: orientation measurements labeled with a bar are not regularized (e.g., $\bar{\delta}$), whereas the orientation measurements without bar (e.g., δ) are already regularized. The algorithm, that will be referred to as *dynamic regularization*, first selects a sub-graph $\mathcal{G}[0]$ of $\mathcal{G}$, that contains only edges corresponding to odometric constraints of $\mathcal{G}$. Let us call $\delta[0]$ the orientation measurements in this initial graph, with $\delta[0] = [\delta_1 \; \delta_2 \; \ldots \; \delta_n]^{\top}$ (according to Section III-E the odometric constraints need no regularization, therefore we simply stack in $\delta[0]$ the given odometric measurements). Then, we want to add a loop closing (chord) at a time, deciding the corresponding regularization term before including it in the graph.At each step of this algorithm we add an edge in the previous graph, obtaining a sequence $\mathcal{G}[1], \mathcal{G}[2], \ldots, \mathcal{G}[\ell]$, such that graph $\mathcal{G}[\ell]$ comprises all constraints (both odometric and loop closings) of the original graph $\mathcal{G}$, i.e., $\mathcal{G}[\ell] = \mathcal{G}$. Let us start from the first iteration: we add the first loop closing to graph $\mathcal{G}[0]$. As mentioned in Section III-E, before including the chord, we have to regularize the corresponding orientation measurement $\bar{\delta}_{n+1}$; we can compute the regularization term for the first chord as follows:

$$\nu_{n+1} = \texttt{round}(c_1^{\top}[1]\bar{\delta}[1]\;,\; 2\pi), \tag{33}$$

where $\bar{\delta}[1] = [\delta[0]^\top \quad \bar{\delta}_{n+1}]^\top$, $c_1(1)$ is the fundamental circuit vector in $\mathcal{G}[1]$ corresponding to chord $n+1$ and $\mathtt{round}(x, y)$ is the operator that rounds x to the closest multiple of y; it is easy to verify that the correction factor in (33) is equal to the one obtained by solving system (26) in the unknown ν_{n+1} after rounding. Applying the correction factor we can update the regularized relative orientation measurements as $\delta[1] = [\delta[0]^\top \quad \bar{\delta}_{n+1} - \nu_{n+1}]^\top$. For computing the second regularization term, static regularization prescribes the following formula:

$$\nu_{n+2} = \mathtt{round}(c_2^\top[2]\bar{\delta}[2] \ , \ 2\pi), \tag{34}$$

where $c_2(2)$ is the circuit vector associated to the second chord and $\bar{\delta}[2] = [\delta[1]^\top \quad \bar{\delta}_{n+2}]^\top$; however, according to the results of Section IV-A, after including the first chord in the graph $\mathcal{G}[0]$, we can compute an estimate for the orientations $\hat{\theta}[1]$. Moreover, thanks to the availability of $\hat{\theta}[1]$, we can also compute a more accurate estimate of the relative orientation measurements in the graph $\mathcal{G}[1]$: $\delta^*[1] = A^\top[1] \cdot \hat{\theta}[1]$, where $A^\top[1]$ is the reduced incidence matrix of $\mathcal{G}[1]$. Therefore, when inserting the second chord we can use a more accurate estimate for computing the regularization terms:

$$\nu_{n+2} = \mathtt{round}(c_2^\top[2]\bar{\delta}^*[2] \ , \ 2\pi), \tag{35}$$

where $\bar{\delta}^*[2] = [(\delta^*[1])^\top \quad \delta_{n+2}^{\,-}]^\top$. In general, we can use the following equation, for computing the regularization term of the chord $(i+1)$, with $i = 1, 2, \ldots, \ell - 1$:

$$\nu_{n+i+1} = \mathtt{round}(c_{i+1}^\top[i+1]\bar{\delta}^*[i+1] \ , \ 2\pi),$$

where $\bar{\delta}^*[i+1] = [A^\top[i] \cdot \hat{\theta}[i] \quad \bar{\delta}_{n+i+1}]^\top$.
Therefore, dynamic regularization exploits the following ideas:

- after including a chord we can update the orientation estimate according to the recursive procedure of Section IV-A;
- the updated orientation estimate allows to compute more accurate relative orientation measurements, that can be used in place of the original orientation measurements to compute the regularization terms to be assigned to the following chords.

In the experimental section we will verify that such approach is able to tackle noisy scenarios in which the state-of-the-art (iterative) techniques may fail. Notice that adopting this incremental approach (*dynamic regularization*) provides at the same time the regularization terms and the solution to phase 1 of Procedure 1, since, by definition, $\hat{\theta}[\ell] = \hat{\theta}$.

V. Experiments and Discussion

In this section we present an experimental evaluation of our approach. Section V-A describes a practical implementation of the approach, whereas in Sections V-B and V-C we report results on accuracy and computational effort.

A. Implementation

We now discuss a practical implementation of our technique that clarifies the previous derivation and is crucial in assuring the computational advantage of the proposed approach. First of all, let us recall that our solution of the batch SLAM problem from real data requires three sequential steps:

1) *Information synthesis*: odometric and loop closing constraints are extracted from real sensor readings;
2) *Regularization*: regularization terms are added to loop closing constraints for making the orientation measurements consistent;
3) *Linear Estimation*: Procedure 1 is applied to solve the graph embedding problem.

As mentioned before, the information synthesis phase 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 *static regularization*

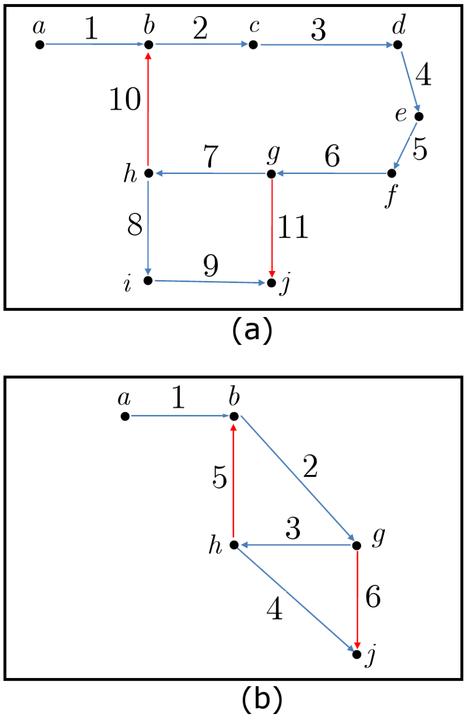


Fig. 2. (a) Example of Pose graph and (b) its *topological reduction*; odometric constraints are shown in blue, whereas loop closing constraints are drawn in red.

is described in Section III-E and the pseudo-code is reported in Algorithm 1. We now briefly provide an example of construction of the cycle basis matrix (24), referred to the graph in Figure 2(a). The odometric constraints are reported as blue segments, whereas the loop closing constraints are in red. Let us number the edges as reported in Figure 2(a). The corresponding relative orientation measurements are $\bar{\delta} = [\bar{\delta}_1 \ \bar{\delta}_2 \ \ldots \ \bar{\delta}_{10} \ \bar{\delta}_{11}]^\top$. The matrix $\mathcal{C}$ has a number of rows equal to the number of loop closing constraints; in the example we have two loop closings. The first row of $\mathcal{C}$ describes the fundamental circuit corresponding to the first loop closing (edge 10), connecting nodes h and b. Therefore the entry in position 10 of vector c_1 is 1 and the other nonzero elements in the circuit vector corresponds to the odometric constraints connecting h and b: the entries corresponding to the odometric edges will be $+1$ if they traverse the loop with the same orientation of the loop closing edge (h, b), otherwise they will be filled with -1. Accordingly, $c_1 = [0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ | \ 1 \ 0]^\top$. Repeating the same procedure for the second loop closing (edge 11), we obtain $c_2 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ -1 \ -1 \ | \ 0 \ 1]^\top$; the cycle basis matrix is $\mathcal{C} = [c_1 \ c_2]^\top$. This procedure for filling in matrix $\mathcal{C}$ corresponds to step 3 in Algorithm 1. Then, the regularization terms can be computed as $\nu_1 = \texttt{round}(\hat{\nu}_1 \ , \ 2\pi)$ and $\nu_2 = \texttt{round}(\hat{\nu}_2 \ , \ 2\pi)$, where $\hat{\nu}_1 = c_1^\top \bar{\delta} = \bar{\delta}_{10} + \sum_{k=2}^{7} \bar{\delta}_k$, and $\hat{\nu}_2 = c_2^\top \bar{\delta} = \bar{\delta}_{11} - \sum_{k=7}^{9} \bar{\delta}_k$ (steps $4-5$ in Algorithm 1). Finally the regularization terms can be used to correct the orientation measurements (step 6 in Algorithm 1).

Algorithm 1 (Static Regularization):

```
1: Input: Incidence matrix $A$, non-regularized orientation measurements $\bar{\delta}$ (of the original graph or of its topological reduction)
2: Output: Regularized measurements $\delta$
3: $\mathcal{C}$ = build_cycle_basis_matrix($A$);
4: $\hat{\nu} = [\mathbf{0}_n^\top \ \ (\mathcal{C}\bar{\delta})^\top]^\top$;
```

```
$\nu = \texttt{round}(\hat{\nu}\ ,\ 2\pi)$;
$\delta = \bar{\delta} - \nu$.
```

Although the construction of matrix $\mathcal{C}$ requires no complex operation, we can further simplify the construction of the cycle basis matrix by observing that the regularization can be performed on a *reduced graph*, comprising only the nodes involved in at least one loop closing constraint. The reduced graph that we shall call *topological reduction* is reported in Figure 2(b). The relative orientation measurements of the topological reduction are $\bar{\delta}^r = [\bar{\delta}^r_1\ \bar{\delta}^r_2\ \dots\ \bar{\delta}^r_6]^\top = [\bar{\delta}_1\ (\sum_{k=2}^{6} \bar{\delta}_k)\ \bar{\delta}_7\ (\sum_{k=8}^{9} \bar{\delta}_k)\ \bar{\delta}_{10}\ \bar{\delta}_{11}]^\top$. The cycle basis matrix is $\mathcal{C}^r = [c^r_1\ c^r_2]^\top$, with $c^r_1 = [0\ 1\ 1\ 0 \mid 1\ 0]^\top$ and $c^r_2 = [0\ 0\ -1\ -1 \mid 0\ 1]^\top$. Then, with simple matrix manipulation it is easy to verify that the regularization vector ν is the same for both graphs, i.e., $\hat{\nu} = \mathcal{C}\bar{\delta} = \mathcal{C}^r\bar{\delta}^r$.

In Algorithm 2 we report the algorithm for *dynamic regularization*, that if necessary can mitigate the orientation wraparound problem (see Section IV-B). The algorithm takes as input the incidence matrix A, the non-regularized orientation measurements $\bar{\delta}$, and the corresponding information matrix $\Omega_\delta \doteq P_\delta^{-1}$ (of the original graph or of its *topological reduction*). As described in Section IV-B the algorithm starts by considering a graph $\mathcal{G}[0]$ containing only odometric constraints. The function `odo_constraints` in Algorithm 2 selects $A[0], \Omega_\delta[0], \delta[0]$, from the overall graph $\mathcal{G}$. Notice that the structure of the matrix $A[0]$ simplifies the computation of the inverse at line 4 of Algorithm 2, see [10]; then, the routine `build_circuit` constructs the circuit vector associated with the i-th chord in the graph $\mathcal{G}[i]$. The function `update` applies equations (31) - (32) for including the edge $(n+i)$ (i-th loop closing) in the estimate. Then, we add a loop closing at a time, deciding incrementally the corresponding regularization term (lines $9-11$). It may be convenient to include the loop closing constraints from the one corresponding to the smallest loop to the one of the biggest loops, for further reducing the risk of selecting incorrect regularization terms; however in all the tests we performed such strategy was not necessary.

Algorithm 2 (Dynamic Regularization):

```
1:  ***Input:*** Incidence matrix $A$, non-regularized orientation measurements $\bar{\delta}$, and corresponding information matrix $\Omega_\delta$ (of the original graph or of its *topological reduction*)
2:  ***Output:*** Regularized measurements $\delta$, suboptimal orientation estimate $\hat{\theta}$
3:  $[A[0], \Omega_\delta[0], \delta[0]] = \texttt{odo\_constraints}(A, \Omega_\delta, \bar{\delta})$;
4:  $P_{\hat{\theta}}[0] = \left(A[0]\ \ \Omega_\delta[0]\ \ A^\top[0]\right)^{-1}$;
5:  $\hat{\theta}[0] = P_{\hat{\theta}}[0]\ A[0]\ \Omega_\delta[0]\ \delta[0]$;
6:  for $i = 1 : \ell$ (i.e., for each loop closing)
7:      $\bar{\delta}^*[i] = \left[\left(A^\top[i-1] \cdot \hat{\theta}[i-1]\right)^\top\ \ \bar{\delta}_{n+i}\right]^\top$;
8:      $c_i[i] = \texttt{build\_circuit}(A)$;
9:      $\hat{\nu}_i = c_i[i]\ \bar{\delta}^*[i]$;
10:     $\nu_i = \texttt{round}(\hat{\nu}_i\ ,\ 2\pi)$;
11:     $\delta_{n+i} = \bar{\delta}_{n+i} - \nu_i$;
12:     $\left[\hat{\theta}[i], P_{\hat{\theta}}[i], A[i]\right] = \texttt{update}(\hat{\theta}[i-1], P_{\hat{\theta}}[i-1], \delta_{n+i}, \Omega_{\delta_{n+i}})$;
13: end
```

After performing regularization (static or dynamic), it is possible to apply Procedure 1. Observing Procedure 1, one may draw the following conclusions: the procedure requires to compute $P_{\hat{\theta}} = (AP_\delta^{-1}A^\top)^{-1}$, to be included in the covariance propagation of phase 2, and P_z^{-1}, which is used in equation (18). In the rest of this section we will show that a suitable implementation of the approach requires no matrix inversion at all. The key insight is that some steps of Procedure 1 can be performed in *information form*, instead of using the corresponding covariance matrices. The pseudocode of Procedure 1 is summarized in Algorithm 3. In the algorithm, function $\texttt{linear\_solver}(\texttt{A}, \texttt{b})$ contains a standard linear equation solver of system $Ax = b$, whereas the content of the routines for building the involved matrices (`build_R`, `build_J`) can be easily derived from Section III.

Algorithm 3 (Linear Estimation):

```
Input: Incidence matrix $A$, regularized orientation measurements $\delta$, information matrix of the relative orientation measurements $\Omega_\delta$, relative position measurements $\Delta^l$, information matrix of relative position measurements $\Omega_{\Delta^l}$
Output: Linear approximation $p^*$
$\Omega_{\hat\theta} = A\Omega_\delta A^\top$;
$\hat\theta = \texttt{linear\_solver}(\Omega_{\hat\theta}\ \ ,\ \ A\Omega_\delta\delta)$
$\hat R = \texttt{build\_R}(\hat\theta)$;
$\Omega_{\Delta^g} = \hat R \Omega_{\Delta^l} \hat R^\top$;
$J = \texttt{build\_J}(\hat\theta, \Delta^l)$;
$\Omega_z = \begin{bmatrix} \Omega_{\Delta^g} & -\Omega_{\Delta^g} J \\ -J\Omega_{\Delta^g} & \Omega_{\hat\theta} + J^\top \Omega_{\Delta^g} J \end{bmatrix}$;
$A_2 = A \otimes \mathbf{I}_2$;
$B = \begin{bmatrix} A_2 & \mathbf{0}_{2n\times n} \\ \mathbf{0}_{n\times 2m} & \mathbf{I}_n \end{bmatrix}$;
$z = [(\hat R \Delta^l)^\top\ \hat\theta^\top]^\top$;
$\Omega_{p^*} = B\Omega_z B^\top$;
$p^* = \texttt{linear\_solver}(\Omega_{p^*}\ \ ,\ \ B\Omega_z z)$.
```

B. Experimental Results: Accuracy

In this section we show an application of the proposed methodology on some publicly available datasets [19], [25]. We consider the situation in which a mobile robot is equipped with wheel encoders and a laser range finder and uses such sensors for performing the *information synthesis* phase, i.e., for computing the constraints in the pose graph. In our experiments, the odometric constraints were obtained by refining wheel odometry measurements with a scan matching algorithm [24]. The loop closing constraints were instead selected from the relations available at [25].

Let us start by considering the *Intel Research Lab* dataset [25]. The number of odometric and loop closing constraints is reported in Table I. In Figure 3 we show the odometric trajectory of the robot (solid line) obtained from wheel encoders estimates and scan matching. The figure also shows, as dotted lines, the edges corresponding to loop closing constraints. The scan matching algorithm is only able to enforce local consistency by aligning the laser readings acquired at subsequent poses, thus failing in producing a globally consistent map. Once the relative pose information $(\bar\delta, \Delta^l)$ is available, we perform a *static regularization* of orientation measurements, obtaining δ. In Table I it is shown the number of loop closing constraints for which a correction factor $2k\pi, k \in \mathbb{Z}\backslash\{0\}$ was needed. Measurements covariance is in the form $P_{ij} = \text{diag}(P_{\Delta^l_{ij}}, P_{\delta_{ij}})$, and the corresponding uncertainties are assumed to be proportional to the respective measurements, e.g. bigger displacements correspond to higher uncertainty. The configuration estimated with Procedure 1 and the corresponding occupancy grid map are reported in Figure 4. The approach allows to correctly embed the pose graph and to accomplish consistent mapping. For a quantitative evaluation we report the values of the SLAM benchmark metrics proposed in [24]: for each constraint we compute the

	Number of nodes	Number of odometric constraints	Number of loop closing constraints	Number of chords needing regularization
INTEL	1228	1227	278	198
MIT	808	807	20	10
FR079	989	988	229	184
CSAIL	1045	1044	128	59
M3500	3500	3499	2099	212

TABLE I

NUMBER OF NODES, ODOMETRIC CONSTRAINTS, LOOP CLOSING CONSTRAINTS, AND REGULARIZED ORIENTATION MEASUREMENTS FOR INTEL RESEARCH LAB DATASET (INTEL), MIT KILLIAN COURT (MIT), FREIBURG BUILDING 079 (FR079), MIT CSAIL BUILDING (CSAIL), AND MANHATTAN WORLD (M3500).

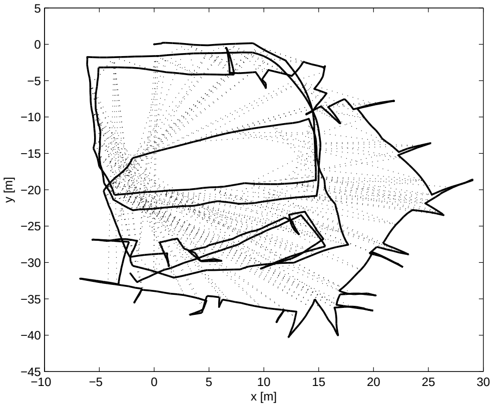
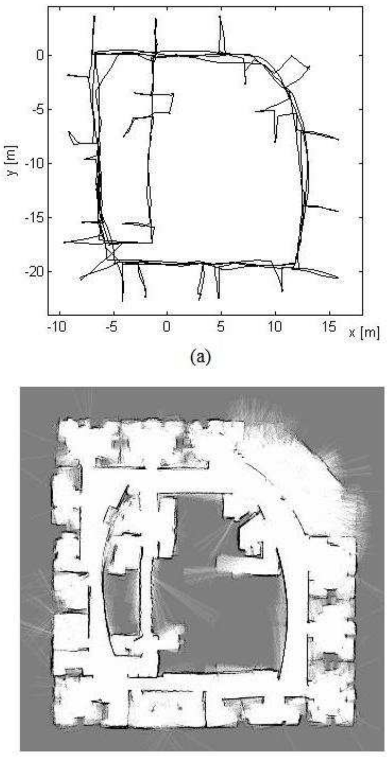


Fig. 3. Intel Research Lab dataset: odometric constraints (solid lines), corresponding to the odometric trajectory corrected with scan matching, and loop closing constraints (dotted lines).

quantity $\eta_{ij} = \xi^*_{ij} - \xi_{ij}$, where ξ^*_{ij} is the relative pose between nodes i and j computed from the estimated configuration, and ξ_{ij} is the corresponding measured quantity. Then, if we distinguish the Cartesian and the angular components in η_{ij}, i.e., $\eta_{ij} = [(\eta^\rho_{ij})^\top \; (\eta^\theta_{ij})]^\top$ the accuracy of the approach is evaluated (i) in terms of Cartesian error $\eta_c = \sqrt{\frac{1}{m}\sum_{(i,j)\in\mathcal{E}} \|\eta^\rho_{ij}\|^2}$, (ii) in terms of angular error $\eta_a = \sqrt{\frac{1}{m}\sum_{(i,j)\in\mathcal{E}} |\eta^\theta_{ij}|^2}$. In Table II we show the values of the *constraint satisfaction metrics*, comparing the proposed solution with (i) a direct iterative method (*Gauss-Newton method*), obtained by solving normal equation (5) until convergence is attained, and (ii) with an indirect iterative method, the *Tree-based netwORk Optimizer* (TORO), available online [15]. The three approaches were tested on the same set of constraints and with the same measurement covariance matrices. For the iterative techniques, the odometric configuration, reported in Figure 3, was assumed as initial guess for optimization. For the Gauss-Newton method, the termination criterion for iterative optimization is the following: the algorithm stops when the norm of the correction vector $\|\delta_p(\tau)\|$ drops below 10^{-1} (recall that $\delta_p(\tau)$ contains thousands of elements, then this threshold is reached when the local correction becomes negligible). For the TORO algorithm, instead, the number of iterations $\bar{\tau}$ is fixed a priori (we repeated the tests with $\bar{\tau} = 100$ and $\bar{\tau} = 1000$). From Table II it is possible to verify that on the INTEL dataset the Gauss-Newton method and the proposed closed-form approximation attain the same minimum. The Gauss-Newton method required 5 iterations for leading the configuration estimate sufficiently close to the minimum. Regarding TORO, the orientation error is comparable with the one of other approaches, while the Cartesian error remains considerably larger also after 1000 iterations.



(a)

(b)

Fig. 4. Intel Research Lab dataset: (a) estimated nodes' configuration, (b) occupancy grid map obtained by associating the corresponding laser scan to each node.

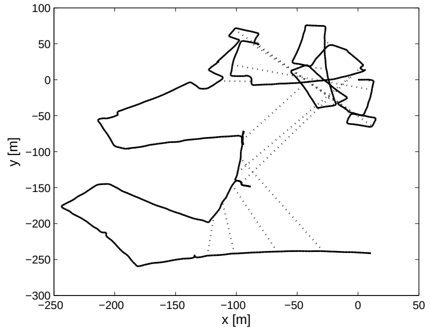


Fig. 5. MIT Killian Court dataset: network configuration after scan matching is applied.

We repeated the previous experiment on the *MIT Killian Court dataset* (*the infinite corridor*). This dataset is challenging, since the robot has to travel hundreds meters in open loop. As in the previous case, the odometric constraints were obtained by aligning the laser readings acquired at subsequent poses. Loop closing constraints were then manually added, as reported in [28]. In Figure 5 we show the network configuration after scan matching is applied. It is possible to notice that the overall structure does not even resemble the actual map, which is reported later and can be compared with the occupancy grid map estimated in [5]. In the same figure we also plot as dotted lines the loop closing constraints. In this case the length of the fundamental circuits and the noise in the orientation measurements prevent to regularize the input measurements with static regularization. However, the use of dynamic regularization is effective in retrieving the correct regularization factors. We report in Table II the values of the *constraint satisfaction metric*, as we did for the previous experiment.We notice that the need of using the iterative

		TORO ($\bar{\tau} = 100$)	TORO ($\bar{\tau} = 1000$)	Gauss-Newton Method	Proposed Approach
INTEL	η_c	$5.15 \cdot 10^{-2}$	$4.31 \cdot 10^{-2}$	$3.03 \cdot 10^{-2}$	$3.03 \cdot 10^{-2}$
	η_a	$1.52 \cdot 10^{-2}$	$1.47 \cdot 10^{-2}$	$1.50 \cdot 10^{-2}$	$1.50 \cdot 10^{-2}$
MIT	η_c	$2.91 \cdot 10^{-1}$	$1.73 \cdot 10^{-1}$	$3.36 \cdot 10^{-1}$ (*)	$1.49 \cdot 10^{-1}$
	η_a	$2.51 \cdot 10^{-2}$	$1.66 \cdot 10^{-2}$	$8.44 \cdot 10^{-2}$ (*)	$1.43 \cdot 10^{-2}$
FR079	η_c	$1.58 \cdot 10^{-2}$	$1.24 \cdot 10^{-2}$	$1.44 \cdot 10^{-2}$	$1.44 \cdot 10^{-2}$
	η_a	$2.22 \cdot 10^{-3}$	$2.04 \cdot 10^{-3}$	$2.02 \cdot 10^{-3}$	$2.02 \cdot 10^{-3}$
CSAIL	η_c	$1.68 \cdot 10^{-2}$	$1.92 \cdot 10^{-2}$	$1.74 \cdot 10^{-2}$	$1.74 \cdot 10^{-2}$
	η_a	$2.13 \cdot 10^{-3}$	$1.73 \cdot 10^{-3}$	$1.73 \cdot 10^{-3}$	$1.73 \cdot 10^{-3}$
M3500	η_c	$1.41 \cdot 10^{-1}$	$1.25 \cdot 10^{-1}$	$1.82 \cdot 10^{-2}$	$1.82 \cdot 10^{-2}$
	η_a	$2.21 \cdot 10^{-2}$	$2.63 \cdot 10^{-2}$	$1.59 \cdot 10^{-2}$	$1.59 \cdot 10^{-2}$

TABLE II

Benchmark error metrics [24] for the Intel Research Lab dataset (INTEL), MIT Killian Court (MIT), Freiburg Building 079 (FR079), MIT CSAIL Building (CSAIL), and Manhattan World (M3500). Translation error η_c is expressed in meters; angular error η_a is in radians. (*) The algorithm converged to a local minimum.

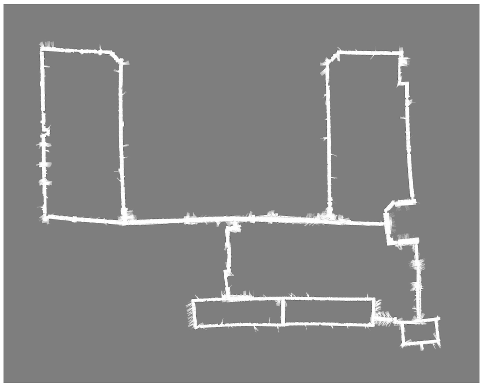


Fig. 6. MIT Killian Court dataset: occupancy grid map corresponding to the configuration estimated with the proposed technique. The size of the overall area shown in figure is approximatively 250 m $\times$ 300 m.

procedure for regularization coincides with the occurrence of the orientation wraparound problem for the Gauss-Newton method: after 24 iterations the Gauss-Newton method converged to a local minimum. The TORO approach, instead, is more resilient to the wraparound problem, as commented in [15], but in this test it turned out to be less accurate than the proposed approach. Figure 6 shows the occupancy grid map, corresponding to the configuration estimated with our approach. The reader can observe that few loop-closing constraints (20 in our test) suffice to correctly recover the structure of the scenario.

We further tested the approach on the following scenarios: (i) *Freiburg Building* 079 (FR079), (ii) *MIT CSAIL Building* (CSAIL), and (iii) the *Manhattan world* (M3500). The information related to the number

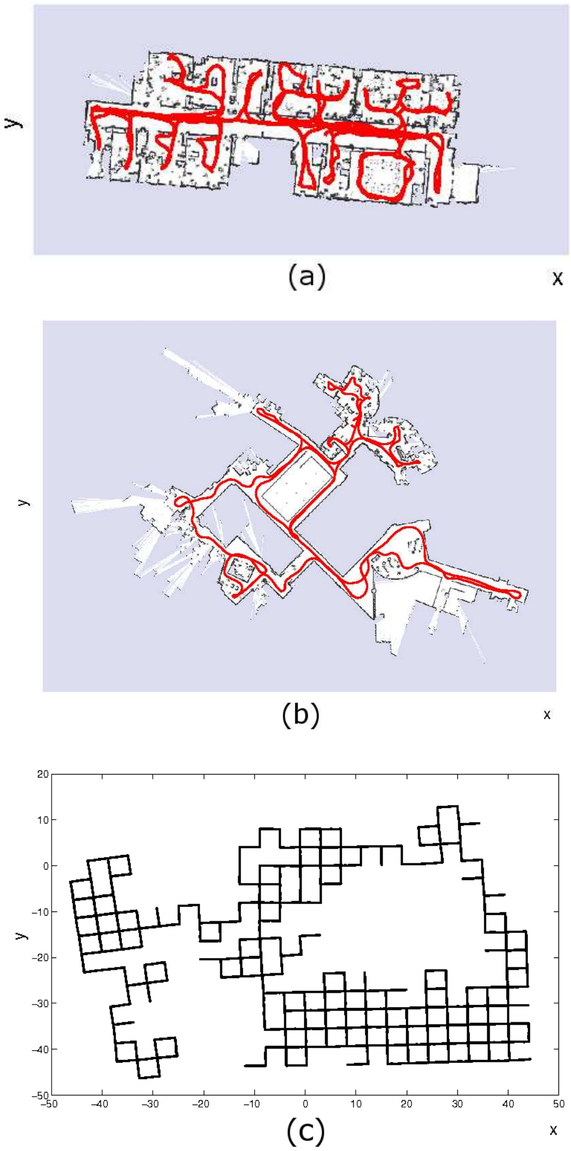


Fig. 7. (a) Freiburg Building 079: configuration estimated with the proposed approach superimposed on the reference map available online, (b) MIT CSAIL Building: configuration estimated with the proposed approach superimposed on the reference map available online, (C) Manhattan World: configuration estimated with the proposed approach.

of nodes and constraints in these scenarios is summarized in Table I. The first two datasets are built on real sensor readings, whereas the Manhattan world is a well-known simulated benchmark scenario [10]. In these cases, we used the relations available at [5] as inputs for the compared algorithms. The results are summarized in Table II: it is possible to notice that the errors attained by the Gauss-Newton method after $3 - 5$ iterations are the same as the ones obtained with the proposed closed-form approximation. In the scenarios FR079 and CSAIL the angular errors of TORO are close to the ones of the proposed approach only after 1000 iterations. Moreover, in some cases, increasing the number of iterations of TORO improves the angular errors, but worsens the Cartesian errors. In the Manhattan world, also after 1000 iterations, the Cartesian errors in TORO remains one order of magnitude larger that the proposed approach. In all these scenarios, it was not necessary to use the incremental procedure for regularization, so we applied the procedure in Algorithm 1.

It is now clear that the proposed approach can perform at least as well as the most accurate, state-of-

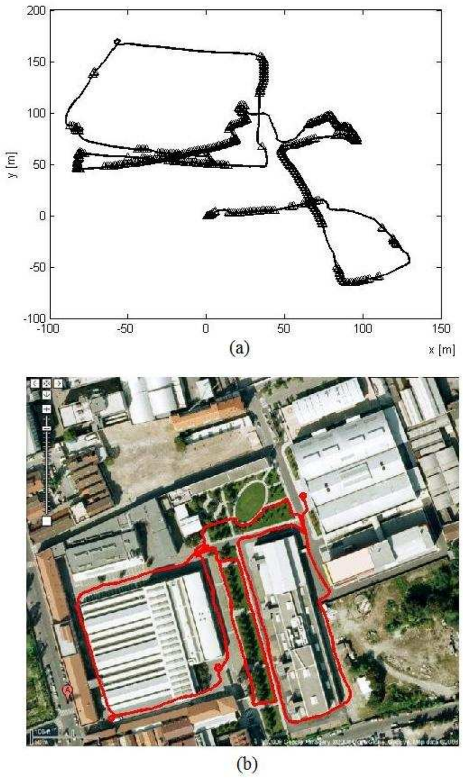


Fig. 8. *Rawseeds* dataset, Bicocca outdoor: (a) odometric trajectory (solid line) with black triangles labeling the poses for which an absolute information from GPS is available, (b) estimated configuration (red) superimposed on the satellite map.

the-art approaches to SLAM, though being a closed-form approximation. We remark that the benchmark metrics considered so far depend on the quality of the constraints, hence the results can be further improved by considering more accurate constraints or a larger set of loop closing constraints.

Remark 1: 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 [15]). □

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 (say, node v_0) and the corresponding edges connect the node that takes the GPS measurement with the node v_0. With straightforward generalizations, we can adapt the proposed approach to deal with such 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 8(a). The configuration, estimated with the proposed approach, is shown in Figure 8(b) superimposed to the satellite map of the outdoor scenario.

C. Experimental Results: Computational Effort

In this section we report an evaluation of the computational effort of the approaches compared in the previous section. We start by observing that the Gauss-Newton method would require solving N linear problems of size $3n$ (in which the number of iterations N may increase arbitrarily, depending on the

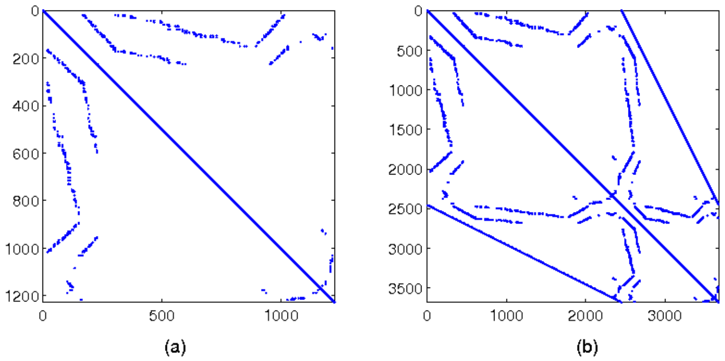


Fig. 9. Sparsity patterns of the matrices involved in phase 1 and 3 of the proposed approach for the INTEL scenario: (a) $\Omega_{\hat{\theta}}$ (see line 4, Algorithm 3); (b) Ω_{p^*} (see line 13, Algorithm 3).

quality of the initial guess); the cost of our approach is due to (i) *regularization*, which only requires (in the static case) to fill in the matrix $\mathcal{C}$ and to perform a (sparse) vector-matrix multiplication, (ii) *linear estimation*, 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. As observed before, the complexity of the computation of robot orientations actually depends on the number of loop closings in the graph (Appendix C), which is usually smaller than the number of nodes in real scenarios (compare with Table I). Therefore the complexity of the proposed method reduces to the solution of a sparse linear system of size $3n$; an example of the sparsity patterns of the involved matrices is reported in Figure 9. The advantage, now, is that the first-order correction provided by the last phase of the proposed approach, refines a suboptimal solution which in practice is already close to a global minimum.

After this discussion we can present the experimental results. The actual implementation of the proposed approach is in Matlab, therefore the results we are going to present are only meant to provide an insight on the effectiveness of approach; we also provide some evidence that *engineering* the code (e.g., writing an optimized code in C++) can speed up the computation of orders of magnitude. In Table III we show the CPU time required for computing the pose graph configuration with the compared approaches. TORO is implemented in C++, whereas the Gauss-Newton method and the proposed approach are implemented in Matlab. The computational effort for TORO is the one returned by the code available online. The time required by the proposed approach, is the sum of the CPU time required for regularization (time for executing the operations in Algorithm 1 or 2) and the time required for computing the linear approximation (operations in Algorithm 3). In particular, we performed the regularization on the topological reduction of the graph (the time needed for computing the topological reduction is also considered in the evaluation of the computational effort); for the *MIT Killian Court* (MIT) the results are referred to *dynamic regularization*, whereas for the other scenarios they are referred to *static regularization*. Time is expressed in seconds (averaged over 10 tests) and the computer used for the evaluation has a processor *INTEL Core 2 Quad*, 2.83 GHz.

	TORO ($\bar{\tau} = 100$)	TORO ($\bar{\tau} = 1000$)	Gauss-Newton Method	Proposed Approach
INTEL	0.63	6.32	2.31 (5)	0.50
MIT	0.33	3.34	39.41 (24) $(^*)$	0.19
FR079	0.50	5.06	0.92 (3)	0.33
CSAIL	0.48	4.79	0.92 (3)	0.30
M3500	2.58	25.87	17.65 (5)	10.50

TABLE III
COMPUTATIONAL EFFORT FOR THE COMPARED TECHNIQUES FOR EACH BENCHMARK SCENARIO. TIME IS EXPRESSED IN SECONDS. FOR THE GAUSS-NEWTON METHOD WE REPORT THE NUMBER OF ITERATION IN PARENTHESIS. $(^*)$ THE ALGORITHM CONVERGED TO A LOCAL MINIMUM.

Let us consider the first four datasets (INTEL, MIT, FR079, CSAIL). Table III shows that in all these datasets, the Matlab implementation is faster than TORO (100 iterations) implementation: most of the operations reported in Algorithms 1 only requires to fill in matrices and the only expensive computation regards the solution of the two linear systems in Algorithm 3, which can be computed efficiently using sparse matrix manipulation in Matlab. Although we observed that in most cases the accuracy of TORO is comparable with the proposed approach only after 1000 iterations, the computational effort for performing such a large number of iterations is undesirable for practical implementations (the computational effort in TORO scales almost linearly with the number of iterations). The Gauss-Newton method is reported only for comparison and confirms the observation reported at the beginning of this section: when the Gauss-Newton method is able to converge to the optimal configuration, its computational time is N times larger than the proposed approach, where N denotes the number of iterations (reported in parenthesis in Table III). Observing the last scenario (M3500) one may argue that the approach has scalability issues. Let us investigate this issue analyzing the computational effort of each phase of the proposed approach. Results are shown in Table IV. The table contains the measured time for performing regularization and for applying Procedure 1 (the CPU time reported in the last column of Table III is the sum of the last two columns of Table IV). The results of the scenario M3500 reveal that most of the time is spent in performing the regularization. This is not surprising since the Manhattan world contains a large number of edges and the `for` cycle that fills in the matrix $\mathcal{C}$ is slow in Matlab. If we exclude the insertion of the elements in the matrix $\mathcal{C}$ from the evaluation, the computational time of regularization on the original graph becomes negligible (parenthesis in the last row, first column of Table IV). Analogously, if we measure the time required for regularization on the topological reduction, excluding the computation of $\mathcal{C}$ (but including the graph reduction), we reduce the computational effort by two orders of magnitude. Regarding the effort for computing Algorithm 3, we observe that, in Matlab, after the operations of lines 5 (definition of B), 7 (definition of J), and 10 (definition of B) of Algorithm 3, we use the command `sparse` for exploiting the structure of the matrices. This command only requires to select the nonzero elements of the corresponding matrices; however, due to the size of the involved matrices, the computational effort of this operation is not negligible in Matlab. We report in parenthesis (last row, last column of Table IV) the CPU time obtained disregarding the three `sparse` commands in Matlab.

As a final observation we remark that the convenience in performing regularization on the topological reduction of the graph is evident in the cases in which the number of loops in the graph is small when compared with the number of nodes; for instance, the advantage is clear in the MIT scenario (Table IV).

	Regularization on Original Graph	Regularization on Topological Reduction	Linear Approximation
INTEL	0.13	0.07	0.43
MIT	0.90	0.05	0.14
FR079	0.06	0.06	0.27
CSAIL	0.02	0.04	0.26
M3500	7.90 (0.02)	5.81 (0.25)	4.68 (1.30)

TABLE IV

Computational effort for each phase of the proposed approach: regularization (on the original graph and on the topological reduction) and linear estimation (Algorithm 3). Time is expressed in seconds.

VI. Conclusion

The contribution of this work is twofold: we propose a linear approximation for the batch SLAM problem, under the assumption that the relative position and the relative orientation measurements are independent. The procedure for computing such approximation requires no initial guess and is formally demonstrated to admit solution when applied to the embedding of the pose graph. It is possible to consider the proposed approach as a linear initialization method for iterative optimization or as a stand-alone tool. Empirical evidence suggests that the approximation is accurate in practice, and, on common SLAM scenarios, it is able to attain the same minimum of a direct iterative approach at a fraction of the computational cost. As a by-product of our result we also show that (i) some steps of the computation of the linear approximation can be solved in incremental way, possibly performed online during robot operation; (ii) if convenient, some steps of the approach can be computed using expressions whose complexity depends on the number of loop-closings in the pose graph. The second contribution regards the orientation wraparound problem. We show that, in the proposed formulation, the wraparound problem has a clear meaning (incapability of discerning the number of 2π turns performed between two place-revisiting episodes); moreover, we propose an approach for alleviating this issue. Future work includes the investigation of possible extensions of the proposed setup to 3D scenarios, in which the first phase of the procedure can no longer be formulated in a linear fashion.

Appendix A

In this appendix we detail the solution to the convex optimization problem (22) obtained by taking a linear approximation of the residual errors in (8) around the suboptimal solution $\hat{p}$. Computing the first-order derivatives of the cost function with respect to δ_θ and δ_ρ and imposing them to be zero we obtain the following linear system:

$$\begin{cases} \frac{\partial f}{\partial \delta_\rho} = 2A_2 P_{\Delta^g}^{-1}\left(A_2^\top \hat{\rho} + A_2^\top \delta_\rho - \hat{R}\Delta^l - J\delta_\theta\right) = \mathbf{0}_{2n} \\ \frac{\partial f}{\partial \delta_\theta} = 2AP_\delta^{-1}\left(A^\top \hat{\theta} + A^\top \delta_\theta - \delta\right) + \\ -2J^\top P_{\Delta^g}^{-1}\left(A_2^\top \hat{\rho} + A_2^\top \delta_\rho - \hat{R}\Delta^l - J\delta_\theta\right) = \mathbf{0}_n \end{cases} . \tag{36}$$

From the first equation, substituting the expression of $\hat{\rho}$ (17), we can rewrite δ_ρ in function of δ_θ:

$$\delta_\rho = P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} J\delta_\theta. \tag{37}$$

Substituting the expressions (15) and (37) in the second equation of (36) we can obtain the value of δ_θ which annihilates the first derivative of the cost function (22):

$$\begin{aligned}\delta_\theta = \Big(P_{\hat{\theta}}^{-1} + J^\top P_{\Delta^g}^{-1} J - J^\top P_{\Delta^g}^{-1} A_2^\top P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} J\Big)^{-1} \\ J^\top P_{\Delta^g}^{-1} \left(A_2^\top P_{\hat{\rho}} A_2 P_{\Delta^g}^{-1} - \mathbf{I}_{2m}\right) \hat{R}\Delta^l.\end{aligned} \tag{38}$$

It is now easy to recognize the first matrix to be P_{p^*}, see (19), hence the previous expression coincides with the one reported in (23). Finally if we substitute again (38) in (37) we can also compute δ_ρ, obtaining the desired result (23).

Appendix B

In this appendix we extend the derivation of Section IV-A to the case in which both nodes and edges are added to an initial graph $\mathcal{G}[0]$. Let us call $\mathcal{G}[1]$ the augmented graph obtained by adding a new node, say v_{n+1}, and the corresponding incident edges, to the graph $\mathcal{G}[0]$. As before, the objective is to devise a recursive expression for computing the estimate $\hat{\theta}[1]$ (and the covariance $P_{\hat{\theta}}[1]$) given the estimate $\hat{\theta}[0]$ (and $P_{\hat{\theta}}[0]$).

Let us assume that there are $\tilde{m}$ edges incident on the new node; accordingly, the reduced incidence matrix $A[1]$ of graph $\mathcal{G}[1]$ can be obtained from $A[0]$ (reduced incidence matrix of $\mathcal{G}[0]$) by adding a new row and $\tilde{m}$ columns. The *augmented* matrix $A[1]$ assumes the form:

$$A[1] = \begin{bmatrix} A[0] & \tilde{A} \\ \mathbf{0}_m^\top & \tilde{a}^\top \end{bmatrix},$$

where $\tilde{a}$ is a vector in $\{-1,+1\}^{\tilde{m}}$ and $\tilde{A} \in \mathbb{R}^{n\times\tilde{m}}$. Notice that the zeros in the last row of $A[1]$ are due to the fact that we are adding only edges that are incident on the new node. If we want to compute the suboptimal orientation estimate for the *augmented* graph with $n+2$ nodes and $m+\tilde{m}$ edges we can use equation (15):

$$\hat{\theta}[1] = \left(A[1]P_\delta^{-1}[1]A^\top[1]\right)^{-1} A[1]P_\delta^{-1}[1]\delta[1], \tag{39}$$

where $\delta[1] = [\delta_1 \;\; \delta_2 \;\; \dots \;\; \delta_{m+\tilde{m}}]^\top$ (we assume that they are already regularized), and $P_\delta[1]$ is the corresponding covariance matrix. As in Section IV-A, we are interested in showing that the computation of (39) does not require the inversion of $\left(A[1] \;\; P_\delta^{-1}[1] \;\; A^\top[1]\right)$, but can exploit the knowledge of $\hat{\theta}[0]$ and $P_{\hat{\theta}}[0]$.

For this purpose, let us write $\delta[1] = [\delta^\top[0] \;\; \tilde{\delta}^\top]^\top$, where $\delta[0]$ contains the relative orientation measurements corresponding to edges in $\mathcal{G}[0]$ and $\tilde{\delta}$ contains the relative orientation measurements corresponding to edges which are in $\mathcal{G}[1]$ but not in $\mathcal{G}[0]$. Accordingly, we rewrite $P_\delta[1] = \mathrm{diag}(P_\delta[0], \tilde{P}_\delta)$, where $P_\delta[0]$ and $\tilde{P}_\delta$ are the covariance matrices of $\delta[0]$ and $\tilde{\delta}$, respectively. It is now easy to show that the covariance matrix $P_{\hat{\theta}}[1]$ can be written as:

$$\begin{aligned} P_{\hat{\theta}}[1] &= \left\{ \begin{bmatrix} A[0] & \tilde{A} \\ \mathbf{0}_m^\top & \tilde{a}^\top \end{bmatrix} \begin{bmatrix} P_\delta[0] & \mathbf{0}_{m\times\tilde{m}} \\ \mathbf{0}_{m\times\tilde{m}}^\top & \tilde{P}_\delta \end{bmatrix}^{-1} \begin{bmatrix} A[0] & \tilde{A} \\ \mathbf{0}_m^\top & \tilde{a}^\top \end{bmatrix}^\top \right\}^{-1} = \\ &= \begin{bmatrix} A[0]P_\delta^{-1}[0]A^\top[0] + \tilde{A}\tilde{P}_\delta^{-1}\tilde{A}^\top & \tilde{A}\tilde{P}_\delta^{-1}\tilde{a} \\ \tilde{a}^\top\tilde{P}_\delta^{-1}\tilde{A}^\top & \tilde{a}^\top\tilde{P}_\delta^{-1}\tilde{a} \end{bmatrix}^{-1} \doteq \begin{bmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{bmatrix}^{-1}. \end{aligned}$$

Using blockwise inversion rule [18], the previous inverse can be computed as:

$$P_{\hat{\theta}}[1] = \begin{bmatrix} \Omega_{11}^{-1} + \Omega_{11}^{-1}\Omega_{12}P_{22}\Omega_{21}\Omega_{11}^{-1} & -\Omega_{11}^{-1}\Omega_{12}P_{22} \\ -P_{22}\Omega_{21}\Omega_{11}^{-1} & P_{22} \end{bmatrix},$$

with $P_{22} = (\Omega_{22} - \Omega_{21}\Omega_{11}^{-1}\Omega_{12})^{-1}$, being a scalar. It still remains the effort of computing Ω_{11}^{-1}, but we can apply the *matrix inversion lemma* [4], simplifying the computation to:

$$\begin{aligned}\Omega_{11}^{-1} = \left(A[0]P_{\delta}^{-1}[0]A^{\top}[0] + \tilde{A}\tilde{P}_{\delta}^{-1}\tilde{A}^{\top}\right)^{-1} = \left(P_{\hat{\theta}}^{-1}[0] + \tilde{A}\tilde{P}_{\delta}^{-1}\tilde{A}^{\top}\right)^{-1} = \\ = P_{\hat{\theta}}[0] - P_{\hat{\theta}}[0]\tilde{A}\left(\tilde{P}_{\delta} + \tilde{A}^{\top}P_{\hat{\theta}}\tilde{A}\right)^{-1}\tilde{A}^{\top}P_{\hat{\theta}}[0].\end{aligned} \tag{40}$$

Equation (40) requires the inversion of $\left(\tilde{P}_{\delta} + \tilde{A}^{\top}P_{\hat{\theta}}\tilde{A}\right)$ which has a size equal to the number of edges incident on node v_{n+1} (it is a scalar if the node v_{n+1} is connected to the rest of the graph only through an odometric constraint). Therefore, we have just demonstrated that, $P_{\hat{\theta}}[1]$ can be computed recursively from $P_{\hat{\theta}}[0]$, without inverting large matrices. Once the covariance $P_{\hat{\theta}}[1]$ has been obtained, the orientation estimate is computed as $\hat{\theta}[1] = A[1]P_{\delta}^{-1}[1]\delta[1]$.

Appendix C

In this appendix we recall some results from linear estimation theory applied to multi agent localization to show that the suboptimal orientation estimate $\hat{\theta}$ and (if needed) the suboptimal position estimate $\hat{\rho}$ can be computed with a complexity that depends on the number of loop closings in the pose graph. For sake of simplicity let us assume that the relative orientation measurements have already been regularized.

Let us call $\hat{\delta}$ the relative orientation computed from the suboptimal configuration, i.e., $\hat{\delta} = A^{\top}\hat{\theta}$. Note that if we are able to compute $\hat{\delta}$, we can easily obtain $\hat{\theta}$; for instance, consider the graph in Figure 1: if $\hat{\delta} = [\hat{\delta}_{01}\ \hat{\delta}_{12}\ \hat{\delta}_{23}\ \hat{\delta}_{34}\ \hat{\delta}_{21}]^{\top}$ is known, then we can compute the suboptimal nodes' orientation 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 the relative orientation information as $\hat{\delta} = [\hat{\delta}_{odo}^{\top}\ \hat{\delta}_{lc}^{\top}]$ such that $\hat{\delta}_{odo}$ corresponds to the edges of the (odometric) spanning tree (i.e., $\hat{\delta}_{odo} = [\hat{\delta}_{01}\ \hat{\delta}_{12}\ \ldots\ \hat{\delta}_{(n-1,n)}]^{\top}$), and $\hat{\delta}_{lc}$ corresponds to the chords in the graph, then we can compute $\hat{\theta} = V\hat{\delta}$, with $V = [U_{n\times n}\ \mathbf{0}_{n\times m-n}]$ and $U_{n\times n}$ 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 4: Given the regularized orientation measurements δ and the corresponding covariance matrix P_{δ}, it is possible to compute the suboptimal estimate $\hat{\delta}$ for nodes' orientations in a connected pose graph as: as:

$$\hat{\theta} = \begin{cases} V\delta & \text{if } m = n \\ V\hat{\delta}, \quad \text{with} \quad \hat{\delta} \doteq \delta - P_{\delta}\mathcal{C}^{\top}(\mathcal{C}P_{\delta}\mathcal{C}^{\top})^{-1}\mathcal{C}\delta & \text{if } m > n \end{cases},$$

where $\mathcal{C}$ is the cycle basis matrix of the underlying graph.

Proof. The proof is a straightforward consequence of Theorem 1 in [31], and of the equality $\hat{\theta} = V\hat{\delta}$. □

The previous result shows that the complexity of the computation of $\hat{\theta}$ depends on the number of cycles in the graph, i.e., on the size of the matrix $(CP_{\delta}C^{\top})$. Moreover, it confirms that, when the graph has no cycle ($m = n$), no optimization needs be performed.

Although we already observed that the approach proposed in this article does not require the computation of $\hat{\rho}$, we further generalize the previous result to the suboptimal position estimate. The proof of the following result easily follows from [31].

Proposition 5: The suboptimal estimate $\hat{\rho}$ of equation (17), can be alternatively computed as:

$$\hat{\rho} = \begin{cases} V_2\hat{R}\Delta^{l} & \text{if } m = n \\ V_2[\hat{R}\Delta^{l} - P_{\Delta^g}\mathcal{C}_2^{\top}(\mathcal{C}_2P_{\Delta^g}\mathcal{C}_2^{\top})^{-1}\mathcal{C}_2\hat{R}\Delta^{l}] & \text{if } m > n \end{cases},$$

where $\mathcal{C}_2 = \mathcal{C} \otimes \mathbf{I}_2$, $V_2 = V \otimes \mathbf{I}_2$, $\hat{R}$ is build as described in Section III, $P_{\Delta^g} \doteq \hat{R}P_{\Delta^l}\hat{R}^{\top}$, Δ^l are the relative position measurements, and P_{Δ^l} is the corresponding covariance matrix. □

Computing $\hat{\rho}$ may be useful when a rough estimate of pose graph configuration is needed. In practice, the suboptimal configuration $\hat{p} = [\hat{\rho}^\top \;\; \hat{\theta}^\top]^\top$ is globally consistent, although the corresponding errors are remarkably bigger than the estimate p^* computed with the proposed approach.

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. *IEEE 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] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[5] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J.D. Tardós. A comparison of SLAM algorithms based on a graph of relations. In *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, 2009.

[6] C. Cadena, D. Galvez-Lopez, F. Ramos, J.D. Tardós, and J. Neira. Robust place recognition with stereo cameras. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 5182–5189, 2010.

[7] 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 lnternational Conf. on Robotics and Automation*, 2011.

[8] Luca Carlone, Rosario Aragues, Jose Castellanos, and Basilio Bona. A linear approximation for graph-based simultaneous localization and mapping. In *Proc. of Robotics: Science and Systems*.

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

[10] 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*, 2010.

[11] 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*, 2002.

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

[13] 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.

[14] 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*, 2010.

[15] 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.

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

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

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

[19] A. Howard and N. Roy. The robotics data set repository (Radish). *http://radish.sourceforge.net*, 2003.

[20] 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*, 2010.

[21] S. Huang, H. Wang, U. Frese, and G. Dissanayake. On the number of local minima to the point feature based slam problem. In *Proc. of IEEE International Conference on Robotics and Automation*, 2012. to appear.

[22] 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.

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

[24] 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.

[25] 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.

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

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

[28] E. Olson, J.J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2262–2269, 2006.

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

[30] D. Lodi Rizzini. A closed-form constraint networks solver for maximum likelihood mapping. In *Proc. of European Conference on Mobile Robots*, 2010.

[31] W.J. Russell, D.J. Klein, and J.P. Hespanha. Optimal estimation on the graph cycle space. In *American Control Conference*, 2010.

[32] 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*, pages 2262–2269, 2010.

[33] N. Sünderhauf and P. Protzel. Towards a robust back-end for pose graph SLAM. In *Proc. of IEEE International Conference on Robotics and Automation*, 2012. to appear.

[34] 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.