Chapter 3
Distributed Localization

Abstract In this chapter we study the problem of distributed localization, which consists of establishing a common frame and computing the robots' localization relative to this frame. Each robot is capable of measuring the relative pose of its neighboring robots. However, it does not know the poses of far robots, and it can only exchange data with neighbors using the range-limited communication network. The analyzed algorithms have the interesting property that can be executed in a distributed fashion. They allow each robot to recover localization using exclusively local information and local interactions with its neighbors. Besides, they only require each robot to maintain an estimate of its own pose. Thus, the memory load of the algorithm is low compared to methods where each robot must also estimate the poses of any other robot. We analyze two different scenarios and study distributed algorithms for them. In the first scenario each robot measures the noisy planar position and orientation of nearby robots to estimate its own full localization with respect to an anchor node. In the second case, robots take noisy measurements of the relative three-dimensional positions of their neighbors, which is used to estimate their three-dimensional positions with respect to the simultaneously computed centroid reference. When the centroid of the team is selected as common frame, the estimates are more precise than with any anchor selection.

Keywords Localization · Limited communication · Distributed systems · Parallel computation

3.1 Introduction

Multi-robot tasks, such as pattern formation [7, 36] or inter-robot collision avoidance [32], often require the knowledge of the robots' positions in a common reference frame. Typically, robots start at unknown locations, they do not share any common frame, and they can only measure the relative positions of nearby robots. We address the localization problem, which consists of combining these relative measurements to build an estimate of the robots' localization in a common frame.

Several localization algorithms rely on range-only [1, 10, 11], or bearing-only [31] relative measurements of the robots' poses. Other approaches assume that robots

R. Aragues et al., *Parallel and Distributed Map Merging and Localization*,
SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-25886-7_3

measure the full state of their nearby robots. The relative full-pose of a pair of robots can be obtained, for instance, by comparing their local maps [14, 15, 33] and looking for overlapping regions. This approach, known as map alignment, presents a high computational cost and its results depend on the accumulated uncertainty in the local maps. Alternatively, each robot can locally combine several observations to build an estimate of the relative poses. The 2D relative pose can be retrieved from at least five noisy distance measurements and four noisy displacements [38]. Bearing-only measurements can be also used to recover the 2D relative pose in vision systems [20]. The 3D case has also been analyzed for distance and bearing, bearing-only, and distance-only observations [35]. These methods present the benefit that the obtained results do not depend on the uncertainties in the local maps. They also allow the robots to compute their relative poses when there is no overlapping between their maps, or even if they do not actually have a map.

Localization algorithms in networked systems properly combine the previous relative measurements to produce an estimate of the robots' poses. Some distributed algorithms compute both, the positions and orientations but assume that the relative measurements are noise free, e.g., [19] where each robot reaches an agreement on the centroid of the network expressed in its local reference frame. Other methods compute exclusively the robot positions but not their orientations, and consider noisy relative measurements of the robot positions. This latter localization problem can be solved by using linear optimization methods [4, 28]. Although these works do not consider the robots' orientations, they can also be applied to such cases provided that the robots have previously executed an attitude synchronization [25, 30] or a motion coordination [16] strategy to align their orientations. Cooperative localization algorithms [22, 27, 34] do not just compute the network localization once, but also track the robots positions. These algorithms, however, usually assume that an initial guess on the robot poses exists.

Formation control [16, 18, 21, 24] and network localization are related problems. While localization algorithms compute robot positions that satisfy the inter-robot restrictions, in formation control problems the robots actually move to these positions. The goal formation is defined by a set of inter-robot restrictions (range-only, bearing-only, full-positions, or relative poses). Although some works discuss the effects of measurement noises in the final result [16], formation algorithms usually assume that both, the measurements and the inter-robot restrictions are noise free [18, 21, 24]. Thus, additional analysis is necessary in noisy localization scenarios.

Both, formation control and localization problems can be solved up to a rotation and a translation. This ambiguity disappears when the positions of a subset of *anchor* robots is given in some absolute reference frame. The range-only case [1] requires at least three non-collinear anchors for planar scenarios. The density and placement of anchors has an important effect on the accuracy of the solution for the bearing-only case [31]. In the full-position case a single anchor is enough. Its placement influences the accuracy of the final results and it is common to analyze the estimation errors at the robots as a function of their distances to the anchor [6]. However, it is common to assume that the first robot is the anchor placed at the origin of the common reference frame and make the other robots compute their positions relative to the anchor.

In this chapter we focus on network localization methods where robots measure the relative pose of their neighbors. Since these methods do not require the robots to have a map, they can be executed at any time. In particular, we execute it at an initial stage, prior to any exploration taking place. The communication graph during this initial stage must be connected. We consider scenarios with noisy relative measurements. We assume that these measurements are independent, since they are acquired individually by the robots. We do not further discuss cooperative localization algorithms, since in a map merging scenario it is enough for the robots to compute the global frame and their poses once. In addition, we discuss the selection of the common reference frame. We consider the cases that the common frame is one of the robots (anchor-based), and that the common frame is the centroid. Firstly we present a distributed algorithm for planar scenarios. Each agent uses noisy measurements of relative planar poses with respect to other robots to estimate its planar localization relative to an anchor node. After that, we discuss the localization problem for higher dimension scenarios. We present a distributed algorithm that allows each robot to simultaneously compute the centroid of the team and its positions relative to the centroid. We show that when the centroid of the team is selected as the common frame, the estimates are more precise than with any anchor selection.

In order to make the reading easy, along the chapter we use the indices i, j to refer to robots and indices e, e' to refer to edges. An edge e starting at robot i and ending at robot j is represented by $e = (i, j)$. Given a matrix A, the notations $A_{r,s}$ and $[A]_{r,s}$ corresponds to the (r, s) entry of the matrix. We let $\otimes$ be the Kronecker product, $\mathbf{I}_r$ be the identity matrix of size $r \times r$, and $\mathbf{0}_{r\times s}$ be a $r \times s$ matrix with all entries equal to zero. A matrix A defined by blocks A_{ij} is denoted $A = [A_{ij}]$. The operation $A = \text{blkDiag}(B_1, \ldots, B_r)$ returns a matrix A defined by blocks with $A_{ii} = B_i$ and $A_{ij} = \mathbf{0}$ for $i \neq j$.

3.2 Problem Description

The problem addressed in this chapter consists of computing the localization of a network of $n \in \mathbb{N}$ robots from relative measurements. We consider two different scenarios.

In the first scenario, the goal is to compute the planar poses of $n \in \mathbb{N}$ robots $\{\mathbf{p}_1^G, \ldots, \mathbf{p}_n^G\}$ expressed in the global frame G, where $\mathbf{p}_i^G = \left[x_i^G, y_i^G, \theta_i^G\right] \in SE(3)$ for $i \in \{1, \ldots, n\}$, given $m \in \mathbb{N}$ measurements of relative poses between robots. The robots measure the planar pose (position and orientation) of nearby robots expressed on their own reference frame. We let $\mathbf{p}_j^i \in SE(3)$ be the pose of a robot j relative to robot i. This information is represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes $\mathcal{V} = \{1, \ldots, n\}$ are the robots, and $\mathcal{E}$ contains the m relative measurements, $|\mathcal{E}| = m$. There is an edge $e = (i, j) \in \mathcal{E}$ from i to j if robot i has a relative measurement of the state of robot j. We assume that the measurement graph $\mathcal{G}$ is directed and weakly connected, and that a robot i can exchange data with both, its in and out neighbors $\mathcal{N}_i$ so that the associated communication graph is undirected,

$$\mathscr{N}_i = \{j \mid (i, j) \in \mathscr{E} \text{ or } (j, i) \in \mathscr{E}\}.$$

We let $\mathscr{A} \in \{0, 1, -1\}^{n\times m}$ be the *negative* incidence matrix of the measurement graph,

$$\mathscr{A}_{i,e} = \begin{cases} -1 & \text{if } e = (i, j) \\ 1 & \text{if } e = (j, i), \\ 0 & \text{otherwise} \end{cases} \text{for } i \in \{1, \ldots, n\}, e \in \{1, \ldots, m\}, \tag{3.1}$$

and we let $\mathscr{W}_{i,j}$ be the Metropolis weights defined in Eq. (A.3) in Appendix A associated to $\mathscr{G}$. The localization problem consists of estimating the states of the n robots from the relative measurements. Any solution can be determined only up to a rotation and a translation, i.e., several equivalent solutions can be obtained depending on the reference frame selected.

As discussed in [4], one of the robots $a \in \mathscr{V}$, e.g., the first one $a = 1$, can be established as an anchor with state $\mathbf{p}_a^a = \mathbf{0}_{3\times 1}$, and the poses of the non-anchor robots can be expressed relative to the anchor. We call such approaches anchor-based and add the superscript a to their associated variables. We let $\mathscr{V}^a = \mathscr{V} \setminus \{a\}$ be the set of non-anchor nodes and matrix $\mathscr{A}^a \in \{0, 1, -1\}^{n-1\times m}$ be the result of deleting the row associated to node a from $\mathscr{A}$ in Eq. (3.1). This is the case considered in our first scenario, where we address the anchor-based planar localization problem for the case that the relative measurements are noisy. Each edge $e = (i, j) \in \mathscr{E}$ in the relative measurements graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ has associated noisy measurements of the orientation $\mathbf{z}_e^\theta$ and the position $\mathbf{z}_e^{xy}$ of robot j relative to robot i, with associated covariance matrices $\Sigma_{\mathbf{z}_e^\theta}$ and $\Sigma_{\mathbf{z}_e^{xy}}$. We assume that the measurements are independent since they were acquired individually by the robots. The goal is to estimate the robot poses $\hat{\mathbf{p}}_i^a$ of the non-anchor robots $i \in \mathscr{V}^a$ relative to the anchor a from the noisy relative measurements. We assume that the orientations of the robots satisfy $-\pi/2 < \theta_i < \pi/2$ for all $i \in \mathscr{V}$.

In the second scenario, instead of computing planar robot poses, we consider that each robot $i \in \mathscr{V}$ has a $p-$dimensional state $\mathbf{x}_i \in \mathbb{R}^p$, and that the measurement $\mathbf{z}_e \in \mathbb{R}^p$ associated to an edge $e = (i, j) \in \mathscr{E}$ relates the states of robots i and j as follows

$$\mathbf{z}_e = \mathbf{x}_j - \mathbf{x}_i + \mathbf{v}_e,$$

where $\mathbf{v}_e \sim N\left(\mathbf{0}_{p\times p}, \Sigma_{\mathbf{z}_e}\right)$ is a Gaussian additive noise. Thus, we solve a position localization problem, although the proposed method can be alternatively applied for estimating speeds, accelerations, or current times. In addition, this method can be used in a pose localization scenario, provided that the robots have previously executed an attitude synchronization [25, 30] or a motion coordination [16] strategy to align their orientations. We estimate the states $\hat{\mathbf{x}}_i^{cen}$ of the robots $i \in \mathscr{V}$ relative to the centroid of the states, and compare the solution with a classical anchor-based one $\hat{\mathbf{x}}_i^a$. In the following sections we explain in detail the two scenarios.

3.3 Planar Localization from Noisy Measurements

The problem addressed in this section consists of computing the planar localization of $n \in \mathbb{N}$ robots $\{\mathbf{p}_1^a, \ldots, \mathbf{p}_n^a\}$, where $\mathbf{p}_i^a = \left[x_i^a, y_i^a, \theta_i^a\right]$ for $i \in \{1, \ldots, n\}$, relative to an anchor robot a, given $m \in \mathbb{N}$ noisy measurements of relative poses between robots. There is a single anchor node $a \in \mathscr{V}$ which is placed at the pose $\mathbf{p}_a^a = \mathbf{0}_{3\times 1}$. By convention, we let the anchor be the first node, $a = 1$, and denote $\mathscr{V}^a = \mathscr{V} \setminus \{a\}$ the set of non-anchor nodes. Each robot gets noisy measurements of the planar pose (position and orientation) of nearby robots to estimate its localization with respect to an anchor node.

Each edge $e = (i, j) \in \mathscr{E}$ in the relative measurements graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ has associated noisy measurements of the orientation $\mathbf{z}_e^\theta$ and the position $\mathbf{z}_e^{xy}$ of robot j relative to robot i, with associated covariance matrices $\Sigma_{\mathbf{z}_e^\theta}$ and $\Sigma_{\mathbf{z}_e^{xy}}$. We let $\mathbf{z}_\theta \in \mathbb{R}^m$, $\mathbf{z}_{xy} \in \mathbb{R}^{2m}$, $\Sigma_{\mathbf{z}_\theta} \in \mathbb{R}^{m\times m}$ and $\Sigma_{\mathbf{z}_{xy}} \in \mathbb{R}^{2m\times 2m}$ contain information of the m measurements,

$$\begin{aligned} \mathbf{z}_\theta &= (\mathbf{z}_1^\theta, \ldots, \mathbf{z}_m^\theta)^T, & \mathbf{z}_{xy} &= ((\mathbf{z}_1^{xy})^T, \ldots, (\mathbf{z}_m^{xy})^T)^T, \\ \Sigma_{\mathbf{z}_\theta} &= \mathrm{Diag}(\Sigma_{\mathbf{z}_1^\theta}, \ldots \Sigma_{\mathbf{z}_m^\theta}), & \Sigma_{\mathbf{z}_{xy}} &= \mathrm{blkDiag}(\Sigma_{\mathbf{z}_1^{xy}}, \ldots \Sigma_{\mathbf{z}_m^{xy}}). \end{aligned}$$

We assume that the measurements are independent since they were acquired individually by the robots. Thus, the goal is that each robot $i \in \mathscr{V}$ estimates its pose $\hat{\mathbf{p}}_i^a$ relative to this anchor.

This problem is solved by using a three-phases strategy [2]

Phase 1: Compute a suboptimal estimate of the robot orientations $\tilde{\theta}_{\mathscr{V}}^a \in \mathbb{R}^n$ relative to the anchor a for all the robots in $\mathscr{V}$;

Phase 2: Express the position measurements $\mathbf{z}_{xy}$ of the robots in terms of the previously computed orientations;

Phase 3: Compute the estimated poses of the robots $\hat{\mathbf{p}}_{\mathscr{V}}^a = ((\hat{\mathbf{x}}_{\mathscr{V}}^a)^T, (\hat{\theta}_{\mathscr{V}}^a)^T)^T$.

During the rest of the section, we study the method and present a distributed implementation.

3.3.1 Centralized Algorithm

Phase 1

During this first phase, an initial estimate of the robot orientations $\tilde{\theta}_{\mathscr{V}^a} \in \mathbb{R}^{n-1}$ relative to the anchor a is obtained. This estimate is computed based exclusively on the orientation measurements $\mathbf{z}_\theta \in \mathbb{R}^m$ with covariance $\Sigma_{\mathbf{z}_\theta} \in \mathbb{R}^{m\times m}$. When the orientation measurements are considered alone and they belong to $\pm\frac{\pi}{2}$, the estimation problem becomes linear, and the estimated solutions are given by the Weighted Least Squares,

$$\tilde{\theta}^a_{\mathcal{V}^a} = \Sigma_{\tilde{\theta}^a_{\mathcal{V}^a}} \mathscr{A}^a \Sigma^{-1}_{\mathbf{z}_\theta} \mathbf{z}_\theta, \qquad \Sigma_{\tilde{\theta}^a_{\mathcal{V}^a}} = \left(\mathscr{A}^a \Sigma^{-1}_{\mathbf{z}_\theta} (\mathscr{A}^a)^T\right)^{-1}, \tag{3.2}$$

where $\mathscr{A}^a \in \{0, 1, -1\}^{n-1\times m}$ is the result of deleting the row associated to the anchor a from the incidence matrix $\mathscr{A}$ of the measurement graph in Eq. (3.1). Recall that the orientation of the anchor is set to zero, $\tilde{\theta}^a_i = 0$ for $i = a$. We let $\tilde{\theta}^a_{\mathcal{V}} \in \mathbb{R}^n$ and $\Sigma_{\tilde{\theta}^a_{\mathcal{V}}} \mathbb{R}^{n\times n}$ contain the orientation of all the robots in $\mathcal{V}$, including the anchor a,

$$\tilde{\theta}^a_{\mathcal{V}} = (0, (\tilde{\theta}^a_{\mathcal{V}^a})^T)^T, \qquad \Sigma_{\tilde{\theta}^a_{\mathcal{V}}} = \text{Diag}(0, \Sigma_{\tilde{\theta}^a_{\mathcal{V}^a}}). \tag{3.3}$$

Phase 2

Each relative position measurement $\mathbf{z}^{xy}_e$ associated to the edge $e = (i, j)$, was originally expressed in the local coordinates of robot i. During the second phase, these measurements are transformed into a common orientation using the previously computed $\tilde{\theta}^a_{\mathcal{V}}$.

For each edge $e = (i, j) \in \mathscr{E}$ we let $\tilde{R}_e \in \mathbb{R}^{2\times 2}$ and $\tilde{S}_e \in \mathbb{R}^{2\times 2}$ be the following matrices associated to the orientation $\tilde{\theta}_i$ of robot i,

$$\tilde{R}_e = \mathscr{R}(\tilde{\theta}^a_i) = \begin{bmatrix} \cos\tilde{\theta}^a_i & -\sin\tilde{\theta}^a_i \\ \sin\tilde{\theta}^a_i & \cos\tilde{\theta}^a_i \end{bmatrix}, \quad \tilde{S}_e = \mathscr{S}(\tilde{\theta}^a_i) = \begin{bmatrix} -\sin\tilde{\theta}^a_i & \cos\tilde{\theta}^a_i \\ -\cos\tilde{\theta}^a_i & -\sin\tilde{\theta}^a_i \end{bmatrix}, \tag{3.4}$$

and let the block diagonal matrix $\tilde{R} \in \mathbb{R}^{2m\times 2m}$ compile information from the m edges,

$$\tilde{R} = \mathscr{R}(\tilde{\theta}^a_{\mathcal{V}}) = \text{blkDiag}(\tilde{R}_1, \ldots, \tilde{R}_m). \tag{3.5}$$

The updated pose measurements in the global coordinates $\mathbf{w} \in \mathbb{R}^{2m+(n-1)}$ and their associated covariance $\Sigma_{\mathbf{w}}$ are

$$\mathbf{w} = \begin{bmatrix} \tilde{\mathbf{z}}_{xy} \\ \tilde{\theta}_{\mathcal{V}^a} \end{bmatrix} = \begin{bmatrix} \tilde{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{z}_{xy} \\ \tilde{\theta}_{\mathcal{V}^a} \end{bmatrix},$$
$$\Sigma_{\mathbf{w}} = \begin{bmatrix} K & J \\ \mathbf{0} & \mathbf{I}_{n-1} \end{bmatrix} \begin{bmatrix} \Sigma_{\mathbf{z}_{xy}} & \mathbf{0} \\ \mathbf{0} & \Sigma_{\tilde{\theta}_{\mathcal{V}^a}} \end{bmatrix} \begin{bmatrix} K^T & \mathbf{0} \\ J^T & \mathbf{I}_{n-1} \end{bmatrix}, \tag{3.6}$$

where $K \in \mathbb{R}^{2m\times 2m}$ and $J \in \mathbb{R}^{2m\times(n-1)}$ are the Jacobians of the transformation with respect to respectively, $\mathbf{z}_{xy}$ and $\tilde{\theta}_{\mathcal{V}^a}$,

$$K = \tilde{R}, \text{ and } \quad J_{e,i} = \tilde{S}_e\, \mathbf{z}^{xy}_e \text{ if } e = (i, j) \text{ for some } j, \text{ and } J_{e,i} = \mathbf{0}_{2\times 1} \text{ otherwise.} \tag{3.7}$$

Phase 3

During the last phase, the positions of the robots $\hat{\mathbf{x}}^a_{\mathcal{V}^a} \in \mathbb{R}^{2(n-1)}$ relative to the anchor node a are computed, and an improved version $\hat{\theta}^a_{\mathcal{V}^a} \in \mathbb{R}^{n-1}$ of the previous orientations $\tilde{\theta}^a_{\mathcal{V}^a}$ is obtained. Let $\hat{\mathbf{p}}^a_{\mathcal{V}^a} \in \mathbb{R}^{3(n-1)}$ contain both, the positions and orientations of the non-anchor robots,

$$\hat{\mathbf{p}}^a_{\mathcal{V}^a} = \begin{bmatrix} \hat{\mathbf{x}}^a_{\mathcal{V}^a} \\ \hat{\theta}^a_{\mathcal{V}^a} \end{bmatrix} = \Sigma_{\hat{\mathbf{p}}^a_{\mathcal{V}^a}} B \Sigma_{\mathbf{w}}^{-1} \mathbf{w}, \qquad \Sigma_{\hat{\mathbf{p}}^a_{\mathcal{V}^a}} = \left(B \Sigma_{\mathbf{w}}^{-1} B^T \right)^{-1}, \tag{3.8}$$

where $B = \text{blkDiag}\left((\mathscr{A}^a \otimes \mathbf{I}_2), \mathbf{I}_{n-1}\right)$, and $\Sigma_{\mathbf{w}}$ and $\mathbf{w}$ are given by (3.6). The estimated poses $\hat{\mathbf{p}}^a_{\mathcal{V}} \in \mathbb{R}^{3n}$ of all the robots in $\mathcal{V}$, including the anchor a, are given by

$$\hat{\mathbf{p}}^a_{\mathcal{V}} = (\mathbf{0}^T_{3\times 1}, (\hat{\mathbf{p}}^a_{\mathcal{V}^a})^T)^T, \qquad \Sigma_{\hat{\mathbf{p}}^a_{\mathcal{V}}} = \text{blkDiag}(\mathbf{0}_{3\times 3}, \Sigma_{\hat{\mathbf{p}}^a_{\mathcal{V}}}). \tag{3.9}$$

Algorithm

Considering the three phases together, the estimated positions $\hat{\mathbf{x}}^a_{\mathcal{V}^a}$ and orientations $\hat{\theta}^a_{\mathcal{V}^a}$ of the non-anchor robots are

$$\begin{aligned} \hat{\mathbf{x}}^a_{\mathcal{V}^a} &= L^{-1}(\mathscr{A}^a \otimes \mathbf{I}_2)\Upsilon_{\tilde{\mathbf{z}}_{xy}} \left(\mathbf{I}_{2m} + J \Sigma_{\hat{\theta}^a_{\mathcal{V}^a}} J^T \Upsilon_{\tilde{\mathbf{z}}_{xy}} E \right) \tilde{R}\, \mathbf{z}_{xy}, \\ \hat{\theta}^a_{\mathcal{V}^a} &= (\mathscr{A}^a \Sigma^{-1}_{\mathbf{z}_\theta} (\mathscr{A}^a)^T)^{-1} \mathscr{A}^a \Sigma^{-1}_{\mathbf{z}_\theta} \mathbf{z}_\theta + \Sigma_{\hat{\theta}^a_{\mathcal{V}^a}} J^T \Upsilon_{\tilde{\mathbf{z}}_{xy}} E\, \tilde{R}\, \mathbf{z}_{xy}, \text{ where} \end{aligned} \tag{3.10}$$

$$\begin{aligned} \Upsilon_{\tilde{\mathbf{z}}_{xy}} &= (\tilde{R} \Sigma_{\mathbf{z}_{xy}} \tilde{R}^T)^{-1}, & E &= (\mathscr{A}^a \otimes \mathbf{I}_2)^T L^{-1} (\mathscr{A}^a \otimes \mathbf{I}_2) \Upsilon_{\tilde{\mathbf{z}}_{xy}} - \mathbf{I}_{2m}, \\ \Sigma_{\hat{\theta}^a_{\mathcal{V}^a}} &= ((\Sigma_{\tilde{\theta}^a_{\mathcal{V}^a}})^{-1} - J^T \Upsilon_{\tilde{\mathbf{z}}_{xy}} E J)^{-1}, & L &= (\mathscr{A}^a \otimes \mathbf{I}_2) \Upsilon_{\tilde{\mathbf{z}}_{xy}} (\mathscr{A}^a \otimes \mathbf{I}_2)^T, \end{aligned} \tag{3.11}$$

and $\hat{\mathbf{p}}^a_{\mathcal{V}}$ is obtained from the previous expressions as in Eq. (3.9). A full development of these expressions can be found in Appendix B. This localization algorithm can also been used for solving the Simultaneously Localization and Mapping problem of single-robot systems building graph maps [12, 13].

3.3.2 *Distributed Algorithm*

From (3.10), it can be seen that the computation of $\hat{\mathbf{x}}^a_{\mathcal{V}}$ and $\hat{\theta}^a_{\mathcal{V}}$ involves matrix inversions and other operations that require the knowledge of the whole system. Although, a priori the proposed localization strategy would require a centralized implementation, in the next sections we show a proposal to carry out the computations in a distributed way.

Phase 1

The initial orientation $\tilde{\theta}^a_{\mathscr{V}}$ in the first phase of the algorithm can be computed in a distributed fashion using the following Jacobi algorithm [4]. Let each robot $i \in \mathscr{V}$ maintain a variable $\tilde{\theta}^a_i(t) \in \mathbb{R}$. The anchor $i = a$ keeps its variable equal to zero for all time steps $t \in \mathbb{N}$,

$$\tilde{\theta}^a_i(0) = 0, \qquad \tilde{\theta}^a_i(t+1) = \tilde{\theta}^a_i(t), \qquad \text{for } i = a. \tag{3.12}$$

Each non-anchor robot $i \in \mathscr{V}^a$ initializes its variable at $t = 0$ with any value $\tilde{\theta}^a_i(0)$, and updates it at each time step $t \in \mathbb{N}$ by

$$\tilde{\theta}^a_i(t+1) = C_i^{-1} c_i + C_i^{-1} \sum_{e=(i,j)\in\mathscr{E}} (\Sigma_{\mathbf{z}^\theta_e})^{-1} \tilde{\theta}^a_j(t) + C_i^{-1} \sum_{e=(j,i)\in\mathscr{E}} (\Sigma_{\mathbf{z}^\theta_e})^{-1} \tilde{\theta}^a_j(t), \tag{3.13}$$

where

$$\begin{aligned} c_i &= -\sum_{e=(i,j)\in\mathscr{E}} (\Sigma_{\mathbf{z}^\theta_e})^{-1} \mathbf{z}^\theta_e + \sum_{e=(j,i)\in\mathscr{E}} (\Sigma_{\mathbf{z}^\theta_e})^{-1} \mathbf{z}^\theta_e, \\ C_i &= \sum_{e=(i,j)\in\mathscr{E}} (\Sigma_{\mathbf{z}^\theta_e})^{-1} + \sum_{e=(j,i)\in\mathscr{E}} (\Sigma_{\mathbf{z}^\theta_e})^{-1}. \end{aligned} \tag{3.14}$$

The previous expressions are the Jacobi iterations associated to (3.2). Let $\Upsilon_{\tilde{\theta}^a_{\mathscr{V}^a}}$ and $\eta_{\tilde{\theta}^a_{\mathscr{V}^a}}$ be respectively the information matrix and vector of $\tilde{\theta}^a_{\mathscr{V}^a}$,

$$\Upsilon_{\tilde{\theta}^a_{\mathscr{V}^a}} = (\Sigma_{\tilde{\theta}^a_{\mathscr{V}^a}})^{-1} = \mathscr{A}^a \Sigma_{\mathbf{z}_\theta}^{-1} (\mathscr{A}^a)^T, \qquad \eta_{\tilde{\theta}^a_{\mathscr{V}^a}} = \mathscr{A}^a \Sigma_{\mathbf{z}_\theta}^{-1} \mathbf{z}_\theta. \tag{3.15}$$

Let C contain the elements in the diagonal of $\Upsilon_{\tilde{\theta}^a_{\mathscr{V}^a}}$,

$$C = \mathrm{Diag}([\Upsilon_{\tilde{\theta}^a_{\mathscr{V}^a}}]_{2,2}, \ldots, [\Upsilon_{\tilde{\theta}^a_{\mathscr{V}^a}}]_{n,n}),$$

and D be $D = C - \Upsilon_{\tilde{\theta}^a_{\mathscr{V}^a}}$. The first equation in (3.2) can be rewritten as

$$\Upsilon_{\tilde{\theta}^a_{\mathscr{V}^a}} \tilde{\theta}^a_{\mathscr{V}^a} = \eta_{\tilde{\theta}^a_{\mathscr{V}^a}}, \qquad \tilde{\theta}^a_{\mathscr{V}^a} = C^{-1} D \tilde{\theta}^a_{\mathscr{V}^a} + C^{-1} \eta_{\tilde{\theta}^a_{\mathscr{V}^a}}. \tag{3.16}$$

From here, we can write

$$\tilde{\theta}^a_{\mathscr{V}^a}(t+1) = C^{-1} D \tilde{\theta}^a_{\mathscr{V}^a}(t) + C^{-1} \eta_{\tilde{\theta}^a_{\mathscr{V}^a}}, \tag{3.17}$$

initialized at $t = 0$ with $\tilde{\theta}^a_{\mathscr{V}^a}(0)$. By operating with $\mathscr{A}^a \Sigma_{\mathbf{z}_\theta}^{-1} \mathbf{z}_\theta$ and $\mathscr{A}^a \Sigma_{\mathbf{z}_\theta}^{-1} (\mathscr{A}^a)^T$, it can be seen that (3.13) is the ith row of (3.17). The system (3.17) converges to

$\tilde{\theta}^a_{\mathcal{V}^a}$ in Eq. (3.2), and equivalently each $\tilde{\theta}^a_i(t)$ in (3.13) converges to $\tilde{\theta}^a_i$ for $i \in \mathcal{V}^a$, if the spectral radius of $C^{-1}D$ is less than 1,

$$\rho(C^{-1}D) < 1, \tag{3.18}$$

and the anchor variable, $\tilde{\theta}^a_i(t)$ with $i = a$, remains equal to 0 for all the iterations t. The value $\rho(C^{-1}D)$ gives the convergence speed of the system, converging faster for $\rho(C^{-1}D)$ closer to 0. Recalling that $\Sigma_{\mathbf{z}_\theta}$ is a diagonal matrix, then each variable $\tilde{\theta}^a_i(t)$ asymptotically converges to the ith entry $\tilde{\theta}^a_i$ of the vector $\tilde{\theta}^a_{\mathcal{V}^a}$ in (3.2) [4] that would be computed by a centralized system.

Observe that the computations are fully distributed and they exclusively rely on local information. The constants C_i and c_i are computed by each robot $i \in \mathcal{V}^a$ using exclusively the measurements $\mathbf{z}^\theta_e$ and covariances $\Sigma_{\mathbf{z}^\theta_e}$ of its incoming $e = (j, i)$ or outgoing edges $e = (i, j)$. Also the variables $\tilde{\theta}^a_j(t)$ used to update its own $\tilde{\theta}^a_i(t + 1)$ belong to neighboring robots $j \in \mathcal{N}_i$.

Phase 2

Let us assume that the robots have executed $t_{\max}$ iterations of the previous algorithm, and let $\bar{\theta}^a_i$ be their orientation at iteration $t_{\max}$, $\bar{\theta}^a_i = \tilde{\theta}^a_i(t_{\max})$. Then, the second phase of the algorithm is executed to transform the locally expressed measurements $\mathbf{z}_{xy}$ into the measurements expressed in the reference frame of the anchor node $\tilde{\mathbf{z}}_{xy}$. As previously stated, the estimated orientations $\bar{\theta}^a_i$ do not change during this phase (3.6). Let $\bar{R} = \mathscr{R}(\bar{\theta}^a_{\mathcal{V}^a})$ be defined by using the orientations $\bar{\theta}^a_i$ instead of $\tilde{\theta}^a_i$ in (3.5). Since the matrix $\bar{R}$ is block diagonal, each robot $i \in \mathcal{V}$ can locally transform its own local measurements,

$$\bar{\mathbf{z}}^{xy}_e = \bar{R}_e \mathbf{z}^{xy}_e, \text{ for all } e = (i, j) \in \mathcal{E}. \tag{3.19}$$

Since the robots use $\bar{\theta}$ instead of $\tilde{\theta}$, also the updated measurements obtained during the second phase are $\bar{\mathbf{z}}_{xy}$ instead of $\tilde{\mathbf{z}}_{xy}$. This second phase is local and it is executed in a single iteration.

Phase 3

In order to obtain the final estimate $\hat{\mathbf{p}}^a_{\mathcal{V}^a}$, the third step of the algorithm (3.8) apparently requires the knowledge of the covariance matrix $\Sigma_{\mathbf{w}}$, which at the same time, requires the knowledge of $\Sigma_{\tilde{\theta}^a_{\mathcal{V}^a}}$. However, a distributed computation of these matrices cannot be carried out in an efficient way. Here we present a distributed algorithm for computing $\hat{\mathbf{p}}^a_{\mathcal{V}^a}$.

Let each robot $i \in \mathcal{V}$ maintain a variable $\hat{\mathbf{p}}^a_i(t) \in \mathbb{R}^3$, composed of its estimated position $\hat{\mathbf{x}}^a_i(t) \in \mathbb{R}^2$ and orientation $\hat{\theta}^a_i(t) \in \mathbb{R}$, and let $\hat{\mathbf{p}}^a_{\mathcal{V}}(t)$ be the result of putting together the $\mathbf{p}^a_i(t)$ variables for all $i \in \mathcal{V}$. The anchor robot keeps its variable equal to zero for all the iterations,

$$\hat{\mathbf{p}}_i^a(0) = \mathbf{0}_{3\times 1}, \qquad \hat{\mathbf{p}}_i^a(t+1) = \hat{\mathbf{p}}_i^a(t), \qquad \text{for } i = a. \tag{3.20}$$

Each non-anchor robot $i \in \mathscr{V}^a$ initializes its variable at $t = 0$ with any value $\hat{\mathbf{p}}_i^a(0)$ and updates $\hat{\mathbf{p}}_i^a(t)$ at each time step $t \in \mathbb{N}$ by

$$\hat{\mathbf{p}}_i^a(t+1) = \begin{bmatrix} \hat{\mathbf{x}}_i^a(t+1) \\ \hat{\theta}_i^a(t+1) \end{bmatrix} = M_i^{-1}\left(\mathbf{f}_i(\hat{\mathbf{p}}_{\mathscr{V}}^a(t)) + \mathbf{m}_i\right), \tag{3.21}$$

where

$$M_i = \begin{bmatrix} M_1 & M_2 \\ M_3 & M_4 \end{bmatrix}, \; \mathbf{f}_i(\mathbf{p}_{\mathscr{V}}^a(t)) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \; \mathbf{m}_i = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}. \tag{3.22}$$

Let $\Upsilon_{\tilde{\mathbf{z}}_e^{xy}}$ be the block within the matrix $\Upsilon_{\tilde{\mathbf{z}}_{xy}}$ in (3.11) associated to an edge $e = (i, j) \in \mathscr{E}$,

$$\Upsilon_{\tilde{\mathbf{z}}_e^{xy}} = \tilde{R}_e(\Sigma_{\mathbf{z}_e^{xy}})^{-1}(\tilde{R}_e)^T. \tag{3.23}$$

The elements within M_i are

$$\begin{aligned}
M_1 &= \sum_{e=(i,j)\in\mathscr{E}} \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} + \sum_{e=(j,i)\in\mathscr{E}} \Upsilon_{\tilde{\mathbf{z}}_e^{xy}}, \\
M_2 &= \sum_{e=(i,j)\in\mathscr{E}} \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \tilde{S}_e \, \mathbf{z}_e^{xy}, \\
M_3 &= \sum_{e=(i,j)\in\mathscr{E}} (\mathbf{z}_e^{xy})^T (\tilde{S}_e)^T \Upsilon_{\tilde{\mathbf{z}}_e^{xy}}, \\
M_4 &= \sum_{e=(i,j)\in\mathscr{E}} (\mathbf{z}_e^{xy})^T (\tilde{S}_e)^T \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \tilde{S}_e \, \mathbf{z}_e^{xy} + \sum_{e=(i,j)\in\mathscr{E}} (\Sigma_{\mathbf{z}_e^{xy}})^{-1} + \sum_{e=(j,i)\in\mathscr{E}} (\Sigma_{\mathbf{z}_e^{xy}})^{-1}.
\end{aligned} \tag{3.24}$$

The elements within $\mathbf{f}_i(\hat{\mathbf{p}}_{\mathscr{V}}^a(t))$, which is the term depending on the previous estimates $\hat{\mathbf{p}}_{\mathscr{V}}^a(t) = (\hat{\mathbf{x}}_{\mathscr{V}}^a(t)^T, \theta\hat{(t)}^{a}_{\mathscr{V}})^T$, are

$$\begin{aligned}
f_1 &= \sum_{e=(i,j)\in\mathscr{E}} \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \, \hat{\mathbf{x}}_j^a(t) + \sum_{e=(j,i)\in\mathscr{E}} \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \, \hat{\mathbf{x}}_j^a(t) + \sum_{e=(j,i)\in\mathscr{E}} \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \, \tilde{S}_e \, \mathbf{z}_e^{xy} \, \hat{\theta}_j^a(t), \\
f_2 &= \sum_{e=(i,j)\in\mathscr{E}} (\mathbf{z}_e^{xy})^T (\tilde{S}_e)^T \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \hat{\mathbf{x}}_j^a(t) - \sum_{e=(i,j)\in\mathscr{E}} (\Sigma_{\mathbf{z}_e^{xy}})^{-1} \hat{\theta}_j^a(t) - \sum_{e=(j,i)\in\mathscr{E}} (\Sigma_{\mathbf{z}_e^{xy}})^{-1} \hat{\theta}_j^a(t).
\end{aligned} \tag{3.25}$$

Finally, the terms within $\mathbf{m}_i$ are

$$
\begin{aligned}
m_1 = &- \sum_{e=(i,j)\in\mathscr{E}} \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \tilde{\mathbf{z}}_e^{xy} + \sum_{e=(j,i)\in\mathscr{E}} \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \tilde{\mathbf{z}}_e^{xy} \\
&+ \sum_{e=(i,j)\in\mathscr{E}} \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \tilde{S}_e \mathbf{z}_e^{xy} \tilde{\theta}_i^a - \sum_{e=(j,i)\in\mathscr{E}} \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \tilde{S}_e \mathbf{z}_e^{xy} \tilde{\theta}_j^a, \\
m_2 = &- \sum_{e=(i,j)\in\mathscr{E}} (\mathbf{z}_e^{xy})^T (\tilde{S}_e)^T \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \tilde{\mathbf{z}}_e^{xy} + \sum_{e=(i,j)\in\mathscr{E}} (\mathbf{z}_e^{xy})^T (\tilde{S}_e)^T \Upsilon_{\tilde{\mathbf{z}}_e^{xy}} \tilde{S}_e \mathbf{z}_e^{xy} \tilde{\theta}_i^a \\
&- \sum_{e=(i,j)\in\mathscr{E}} (\Sigma_{\mathbf{z}_e^{\theta}})^{-1} \tilde{\theta}_j^a - \sum_{e=(j,i)\in\mathscr{E}} (\Sigma_{\mathbf{z}_e^{\theta}})^{-1} \tilde{\theta}_j^a \\
&+ \sum_{e=(i,j)\in\mathscr{E}} (\Sigma_{\mathbf{z}_e^{\theta}})^{-1} \tilde{\theta}_i^a + \sum_{e=(j,i)\in\mathscr{E}} (\Sigma_{\mathbf{z}_e^{\theta}})^{-1} \tilde{\theta}_i^a.
\end{aligned}
\tag{3.26}
$$

Theorem 5 *The estimates $\hat{\mathbf{p}}_i^a(t)$ computed by each robot $i \in \mathscr{V}$ by the distributed algorithm (3.20)–(3.21) converge to $\hat{\mathbf{p}}_i^a = [(\hat{\mathbf{x}}_i^a)^T \hat{\theta}_i^a]^T$ for connected measurement graphs $\mathscr{G}$ with ring or string structure.*

Proof For the anchor $i = a$, it is true since $\hat{\mathbf{p}}_i^a(t) = \mathbf{0}$ for all the time steps. Now we focus on the non-anchor nodes in $\mathscr{V}^a$. First of all, we show that $\hat{\mathbf{p}}_i^a$ is an equilibrium point of the algorithm (3.21) for all $i \in \mathscr{V}^a$. Let $\Upsilon_{\hat{\mathbf{p}}_{\mathscr{V}^a}^a}$ be the information matrix associated to $\hat{\mathbf{p}}_{\mathscr{V}^a}^a$, i.e., $\Upsilon_{\hat{\mathbf{p}}_{\mathscr{V}^a}^a} = (\Sigma_{\hat{\mathbf{p}}_{\mathscr{V}^a}^a})^{-1}$,

$$
\Upsilon_{\hat{\mathbf{p}}_{\mathscr{V}^a}^a} = \begin{bmatrix} L & -\mathscr{A}^a \Upsilon_{\tilde{\mathbf{z}}_{xy}} J \\ -J^T \Upsilon_{\tilde{\mathbf{z}}_{xy}} (\mathscr{A}^a \otimes \mathbf{I}_2)^T & \mathscr{A}^a \Sigma_{\mathbf{z}_\theta}^{-1} (\mathscr{A}^a)^T + J^T \Upsilon_{\tilde{\mathbf{z}}_{xy}} J \end{bmatrix}, \tag{3.27}
$$

where L and $\Upsilon_{\tilde{\mathbf{z}}_{xy}}$ are given by (3.11). Analyzing the term $B \Sigma_{\mathbf{w}}^{-1}$ in (3.8), it can be seen that

$$
B \Sigma_{\mathbf{w}}^{-1} = \begin{bmatrix} (\mathscr{A}^a \otimes \mathbf{I}_2) \Upsilon_{\tilde{\mathbf{z}}_{xy}} & -(\mathscr{A}^a \otimes \mathbf{I}_2) \Upsilon_{\tilde{\mathbf{z}}_{xy}} J \\ -J^T \Upsilon_{\tilde{\mathbf{z}}_{xy}} & \mathscr{A}^a \Upsilon_{\mathbf{z}_\theta}^{-1} (\mathscr{A}^a \otimes \mathbf{I}_2)^T + J^T \Upsilon_{\tilde{\mathbf{z}}_{xy}} J \end{bmatrix}. \tag{3.28}
$$

If we express the third phase in the following way

$$
\Upsilon_{\hat{\mathbf{p}}_{\mathscr{V}^a}^a} \, \hat{\mathbf{p}}_{\mathscr{V}^a}^a = B \, \Sigma_{\mathbf{w}}^{-1} \, \mathbf{w}, \tag{3.29}
$$

and then we consider the rows associated to robot i, we get

$$
\hat{\mathbf{p}}_i^a = \begin{bmatrix} \hat{\mathbf{x}}_i^a \\ \hat{\theta}_i^a \end{bmatrix} = M_i^{-1} \left(\mathbf{f}_i(\hat{\mathbf{p}}_{\mathscr{V}}^a) + \mathbf{m}_i \right), \tag{3.30}
$$

with M_i, $\mathbf{f}_i(\mathbf{p}_{\mathscr{V}}^a(t))$ and $\mathbf{m}_i$ as in (3.22)–(3.26).

Now we prove that the system is convergent. Let $M = \mathrm{blkDiag}(M_2, \ldots, M_n)$ and $\hat{\mathbf{q}}^a_{\mathcal{V}^a}$ be a permutation of $\hat{\mathbf{p}}^a_{\mathcal{V}^a}$ so that the estimates of each robot appear together, $\hat{\mathbf{q}}^a_{\mathcal{V}^a} = \left[(\hat{\mathbf{x}}^a_2)^T \hat{\theta}^a_2, \ldots, (\hat{\mathbf{x}}^a_n)^T \hat{\theta}^a_n\right]^T$. Equivalently, the permuted version of the information matrix $\Upsilon_{\hat{\mathbf{p}}^a_{\mathcal{V}^a}}$ is $\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}}$. The estimates $\hat{\mathbf{p}}^a_i(t)$ computed by each robot $i \in \mathcal{V}^a$ with the distributed algorithm (3.21) converge to $\hat{\mathbf{p}}^a_i = [(\hat{\mathbf{x}}^a_i)^T \hat{\theta}^a_i]^T$ if $\rho(M^{-1}(M - \Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}})) < 1$, or equivalently if

$$\rho(\mathbf{I} - M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}}) < 1. \tag{3.31}$$

Since $\lambda(\mathbf{I} - M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}}) = 1 - \lambda(M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}})$, then (3.21) converges if $0 < \lambda(M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}}) < 2$. The first part $0 < \lambda(M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}})$ can be easily checked taking into account that both M^{-1} and $\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}}$ are nonsingular, symmetric, positive definite, and that $\lambda(M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}}) \geq \frac{\lambda_{\min}(M^{-1})}{\lambda_{\max}(\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}})}$ [23, Lemma 1]. Since $0 < \frac{\lambda_{\min}(M^{-1})}{\lambda_{\max}(\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}})}$, then $0 < \lambda(M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}})$.

In order to prove the second part, $\lambda(M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}}) < 2$, let us first focus on the structure of the information matrix $\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}}$. This matrix has zeros for the elements associated to non neighboring robots, and thus it is compatible with $\mathrm{adj}(\mathcal{G}) \otimes \mathbf{I}_3$, where $\mathrm{adj}(\mathcal{G})$ is the adjacency matrix of the graph, and $\mathbf{I}_3$ is the 3×3 identity matrix. For ring or string graphs, the adjacency matrix can be reordered grouping the elements around the main diagonal resulting in a matrix that has semi bandwidth $s = 1$, i.e.,

$$\mathrm{adj}(\mathcal{G})_{ij} = 0 \text{ for } |i - j| > s.$$

As a consequence, the information matrix $\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}}$ has block semi bandwidth $s' = 1$, and as stated by [23, Theorem 1],

$$\lambda_{\max}(M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}}) < 2^{s'} = 2.$$

□

Due to the structure of the information matrices, the third phase of the algorithm can be expressed in terms of local information (3.21)–(3.26) and interactions with neighbors, and thus it can be implemented in a distributed fashion. It is observed that the robots actually use $\bar{\theta}^a_{\mathcal{V}}$ instead of $\tilde{\theta}^a_{\mathcal{V}}$ and as a result, the solution obtained is slightly different from the one in the centralized case. We experimentally analyze the effects of these differences later in this chapter.

3.4 Centroid-Based Position Estimation from Noisy Measurements

This section discusses a higher dimensional scenario. We addresses the problem of estimation of position from noisy measurements of the relative positions of neighbors. The method simultaneously estimates the centroid of the network. Each robot in the network obtains its three dimensional position relative to the estimated centroid. The usual approaches to multi-robot localization assume instead that one anchor robot exists in the network, and the other robots positions are estimated with respect to the anchor. We show that the studied centroid-based algorithm converges to the optimal solution, and that such a centroid-based representation produces results that are more accurate than anchor-based ones, irrespective of the selected anchor [3]. In previous sections we denoted $\mathbf{p}_i$ the pose of a robot i. Since in this section we exclusively consider robot positions, for clarity we use a different symbol $\mathbf{x}_i$ for the robots variables.

Consider that each robot $i \in \{1, \dots, n\}$ has a $p-$dimensional state $\mathbf{x}_i \in \mathbb{R}^p$ and it observes the states of a subset of the robots relative to its own state, $\mathbf{x}_j - \mathbf{x}_i$. These states can be positions in cartesian coordinates or, in other situations, orientations, speeds, accelerations, or current times. Each edge $e = (i, j) \in \mathscr{E}$ in the relative measurements graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ represents that robot i has a noisy relative measurement $\mathbf{z}_e \in \mathbb{R}^p$ of the state of robot j,

$$\mathbf{z}_e = \mathbf{x}_j - \mathbf{x}_i + \mathbf{v}_e, \tag{3.32}$$

where $\mathbf{v}_e \sim N\left(\mathbf{0}_{p\times p}, \Sigma_{\mathbf{z}_e}\right)$ is a Gaussian additive noise. We let $\mathbf{z} \in \mathbb{R}^{mp}$ and $\Sigma_{\mathbf{z}} \in \mathbb{R}^{mp\times mp}$ contain the information of the m measurements,

$$\mathbf{z} = (\mathbf{z}_1^T, \dots, \mathbf{z}_m^T)^T, \qquad \Sigma_{\mathbf{z}} = \text{blkDiag}(\Sigma_{\mathbf{z}_1}, \dots, \Sigma_{\mathbf{z}_m}). \tag{3.33}$$

We assume that the measurement graph $\mathscr{G}$ is directed and weakly connected, and that an robot i can exchange data with both its in and out neighbors $\mathscr{N}_i$ so that the associated communication graph is undirected. The estimation from relative measurements problem consists of estimating the states of the n robots from the relative measurements $\mathbf{z}$. Any solution can be determined only up to an additive constant. Conventionally [4] one of the robots $a \in \mathscr{V}$, e.g., the first one $a = 1$, is established as an anchor with state $\hat{\mathbf{x}}_a^a = \mathbf{0}_p$. We call such approaches anchor-based and add the superscript a to their associated variables. The Best Linear Unbiased Estimator of the states $\hat{\mathbf{x}}_{\mathscr{V}^a}^a \in \mathbb{R}^{(n-1)p}$, $\hat{\mathbf{x}}_{\mathscr{V}^a}^a = ((\hat{\mathbf{x}}_2^a)^T, \dots, (\hat{\mathbf{x}}_n^a)^T)^T$, of the non-anchor robots $\mathscr{V}^a = \mathscr{V} \setminus \{a\}$ relative to a are obtained as follows [4],

$$\hat{\mathbf{x}}_{\mathscr{V}^a}^a = \Sigma_{\hat{\mathbf{x}}_{\mathscr{V}^a}^a} \left(\mathscr{A}^a \otimes \mathbf{I}_p\right) \Sigma_{\mathbf{z}}^{-1}\mathbf{z}, \quad \Sigma_{\hat{\mathbf{x}}_{\mathscr{V}^a}^a} = \left((\mathscr{A}^a \otimes \mathbf{I}_p)\Sigma_{\mathbf{z}}^{-1}(\mathscr{A}^a \otimes \mathbf{I}_p)^T\right)^{-1}, \tag{3.34}$$

where $\mathscr{A}^a \in \mathbb{R}^{(n-1)\times m}$ is the incidence matrix of $\mathscr{G}$ as in Eq. (3.1), but without the row associated to the anchor a. From now on, both $\hat{\mathbf{x}}^a_{\mathscr{V}} = (\mathbf{0}_p^T, (\hat{\mathbf{x}}^a_{\mathscr{V}^a})^T)^T$ and $\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}} = \text{blkDiag}\left(\mathbf{0}_{p\times p}, \Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}^a}}\right)$, include the estimated state of the anchor a as well.

3.4.1 Position Estimation Relative to an Anchor

We present first distributed strategies where each robot i iteratively estimates its own position relative to an anchor through local interactions with its neighbors $\mathscr{N}_i$. Among the different existing methods for estimating the states $\hat{\mathbf{x}}^a_{\mathscr{V}}$ relative to an anchor, we present the Jacobi algorithm [4], although other distributed methods such as the Jacobi Overrelaxation [8], or the Overlapping Subgraph Estimator [5] could alternatively be applied. The approach in [28], based on the cycle structure of the graph, could be used as well, although it requires multi-hop communication.

Considering Eq. (3.34), it can be seen that computing $\hat{\mathbf{x}}^a_{\mathscr{V}^a}$ is equivalent to finding a solution to the system $\Upsilon \hat{\mathbf{x}}^a_{\mathscr{V}^a} = \eta$, being η and Υ the information vector and matrix associated to $\hat{\mathbf{x}}^a_{\mathscr{V}^a}$ and $\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}^a}}$,

$$\eta = \left(\mathscr{A}^a \otimes \mathbf{I}_p\right) \Sigma_{\mathbf{z}}^{-1}\mathbf{z}, \qquad \Upsilon = \left(\mathscr{A}^a \otimes \mathbf{I}_p\right) \Sigma_{\mathbf{z}}^{-1} \left(\mathscr{A}^a \otimes \mathbf{I}_p\right)^T. \tag{3.35}$$

This can be iteratively solved with the Jacobi method [8], where the variable $\hat{\mathbf{x}}^a_{\mathscr{V}^a}(t) \in \mathbb{R}^{(n-1)p}$ is initialized with an arbitrary value $\hat{\mathbf{x}}^a_{\mathscr{V}^a}(0)$ and it is updated at each step t with the following rule,

$$\hat{\mathbf{x}}^a_{\mathscr{V}^a}(t+1) = D^{-1}N\hat{\mathbf{x}}^a_{\mathscr{V}^a}(t) + D^{-1}\eta, \tag{3.36}$$

being D, N the following decomposition of $\Upsilon = [\Upsilon_{ij}]$:

$$D = \text{blkDiag}(\Upsilon_{22}, \ldots, \Upsilon_{nn}), \qquad N = D - \Upsilon. \tag{3.37}$$

The previous variable $\hat{\mathbf{x}}^a_{\mathscr{V}^a}(t)$ converges to $\hat{\mathbf{x}}^a_{\mathscr{V}^a}$ if the Jacobi matrix $J = D^{-1}N$ has spectral radius less than or equal to one, $\rho(J) = \rho(D^{-1}N) < 1$. The interest of the Jacobi method is that it can be executed in a distributed fashion when the information matrix Υ is compatible with the graph (if $j \notin \mathscr{N}_i$ then $\Upsilon_{ij} = \Upsilon_{ji} = \mathbf{0}_{p\times p}$), and when in addition the rows of Υ and of η associated to each robot $i \in \mathscr{V}^a$ only depend on data which is local to robot i. Next, the general anchor-based estimation algorithm [4] based on the Jacobi method is presented. It allows each robot $i \in \mathscr{V}$ to iteratively estimate its own $\hat{\mathbf{x}}^a_i$ within $\hat{\mathbf{x}}^a_{\mathscr{V}^a} = ((\hat{\mathbf{x}}^a_2)^T, \ldots, (\hat{\mathbf{x}}^a_n)^T)^T$ in a distributed fashion.

Algorithm 3 Let each robot $i \in \mathscr{V}$ have a variable $\hat{\mathbf{x}}^a_i(t) \in \mathbb{R}^p$ initialized at $t = 0$ with $\hat{\mathbf{x}}^a_i(0) = \mathbf{0}_p$. At each time step t, each robot $i \in \mathscr{V}$ updates $\hat{\mathbf{x}}^a_i(t)$ with

$$\hat{\mathbf{x}}_i^a(t+1) = \sum_{j \in \mathscr{N}_i} M_i \mathscr{B}_{ij} \hat{\mathbf{x}}_j^a(t) + \sum_{e=(j,i)\in\mathscr{E}} M_i \Sigma_{\mathbf{z}_e}^{-1} \mathbf{z}_e - \sum_{e=(i,j)\in\mathscr{E}} M_i \Sigma_{\mathbf{z}_e}^{-1} \mathbf{z}_e, \tag{3.38}$$

where M_i and $\mathscr{B}_{ij}$ are $p \times p$ matrices with $M_i = \mathbf{0}$ for $i = a$, $M_i = (\sum_{j \in \mathscr{N}_i} \mathscr{B}_{ij})^{-1}$ for $i \neq a$, and

$$\mathscr{B}_{ij} = \begin{cases} \Sigma_{\mathbf{z}_e}^{-1} + \Sigma_{\mathbf{z}_{e'}}^{-1} & \text{if } e = (i,j), e' = (j,i) \in \mathscr{E} \\ \Sigma_{\mathbf{z}_e}^{-1} & \text{if } e = (i,j) \in \mathscr{E}, (j,i) \notin \mathscr{E} \\ \Sigma_{\mathbf{z}_e}^{-1} & \text{if } e = (j,i) \in \mathscr{E}, (i,j) \notin \mathscr{E} \end{cases}. \tag{3.39}$$

The convergence of this estimation algorithm has been proved [4, Theorem 1] for connected measurement graphs with independent relative measurements, under the assumption that either

(i) The covariance matrices of the measurements are exactly diagonal; or
(ii) All measurements have exactly the same covariance matrix.

However, we would like the algorithm presented here to be applicable to a wider case of relative noises, in particular to independent noises, with not necessarily diagonal or equal covariance matrices. Next we use results on block matrices [17], see Appendix B, to prove the convergence of the Jacobi algorithm for this more general case.

Theorem 6 *Let the measurement graph $\mathscr{G}$ be weakly connected, $\Sigma_{\mathbf{z}_1}, \ldots, \Sigma_{\mathbf{z}_m}$ be the covariance matrices, not necessarily equal or diagonal, associated to m independent $p-$dimensional measurements, and $\Sigma_{\mathbf{z}}$ be their associated block-diagonal covariance matrix as in Eq.* (3.33)*. Then, the spectral radius of $D^{-1}N$, with D and N computed as in Eqs.* (3.35)–(3.37)*, is less than 1,*

$$\rho(D^{-1}N) < 1. \tag{3.40}$$

Proof In order to prove (3.40) we use the definitions and results in Appendix B. We first analyze the contents of Υ and show that Υ is of class Z_{n-1}^p according to Definition 6 in Appendix B. Then, we use Lemma 4 and Theorem 9 to show that Υ is of class M_{n-1}^p as in Definition 6. Finally, we show that $\Upsilon + \Upsilon^T \in M_{n-1}^p$ and use Theorem 10 to prove (3.40). Note that the subscript $n-1$ used in this proof instead of n comes from the fact that $\Upsilon = [\Upsilon_{ij}]$, with $i, j \in \mathscr{V}^a$ and $|\mathscr{V}^a| = n-1$.

We first analyze the contents of the information matrix Υ given by Eq. (3.35). Each block Υ_{ij} of the information matrix Υ is given by

$$\Upsilon_{ij} = \begin{cases} -\mathscr{B}_{ij} & \text{if } j \in \mathscr{N}_i, j \neq i \\ \mathbf{0} & \text{if } j \notin \mathscr{N}_i, j \neq i \end{cases}, \qquad \text{and } \Upsilon_{ii} = \sum_{j \in \mathscr{N}_i} \mathscr{B}_{ij}, \tag{3.41}$$

for $i, j \in \mathscr{V}^a$, where $\mathscr{B}_{ij}$ is given by Eq. (3.39). Note that $\mathscr{B}_{ij}$ is symmetric and that $\mathscr{B}_{ij} \succ \mathbf{0}$[1] and thus $-\mathscr{B}_{ij} \prec \mathbf{0}$ and symmetric. Therefore, matrix Υ is of class Z^p_{n-1} according to Definition 6.

Now we focus on Lemma 4. We are interested in showing that, given any subset of robots $\mathscr{J} \subset \mathscr{V}^a$, there exists $i \in \mathscr{J}$ such that $\sum_{j \in \mathscr{J}} \Upsilon_{ij} \succ \mathbf{0}$. First we analyze the case $\mathscr{J} = \mathscr{V}^a$. Observe that Υ does not have any rows or columns associated to the anchor robot a, i.e., $\Upsilon = [\Upsilon_{ij}]$ with $i, j \in \mathscr{V}^a$. On the other hand, for each robot i that has the anchor a as a neighbor, $a \in \mathscr{N}_i$, the block Υ_{ii} includes $\mathscr{B}_{ia}$. Therefore, $\sum_{j \in \mathscr{V}^a} \Upsilon_{ij} \succeq \mathbf{0}$ for all $i \in \mathscr{V}^a$, specifically

$$\sum_{j \in \mathscr{V}^a} \Upsilon_{ij} = \mathbf{0} \text{ if } a \notin \mathscr{N}_i, \text{ and } \qquad \sum_{j \in \mathscr{V}^a} \Upsilon_{ij} = \mathscr{B}_{ia} \succ \mathbf{0}, \text{ when } a \in \mathscr{N}_i. \tag{3.42}$$

Since $\mathscr{G}$ is connected, $a \in \mathscr{N}_i$ for at least one robot $i \in \mathscr{V}^a$. Now consider a proper subset $\mathscr{J} \subsetneq \mathscr{V}^a$. Note that for each $i \in \mathscr{J} \subsetneq \mathscr{V}^a$,

$$\sum_{j \in \mathscr{J}} \Upsilon_{ij} = \mathbf{0} \text{ if } \mathscr{N}_i \subseteq \mathscr{J}, \text{ and } \qquad \sum_{j \in \mathscr{J}} \Upsilon_{ij} = \sum_{j \in \mathscr{N}_i \setminus \mathscr{J}} \mathscr{B}_{ij} \succ \mathbf{0}, \text{ otherwise.} \tag{3.43}$$

Since $\mathscr{G}$ is connected, given any proper subset $\mathscr{J} \subsetneq \mathscr{V}^a$ of robots, there is always a robot $i \in \mathscr{J}$ that has at least one neighbor outside $\mathscr{J}$ or that has the anchor a as a neighbor, for which $\sum_{j \in \mathscr{J}} \Upsilon_{ij} \succ \mathbf{0}$. Therefore Lemma 4 holds, and by applying Theorem 9 taking $u_2, \ldots, u_n = 1$ we conclude that matrix $\Upsilon \in M^p_{n-1}$. Since Υ is symmetric, then $\Upsilon + \Upsilon^T \in M^p_{n-1}$, and by [17, Theorem 4.7] we conclude that $\rho(D^{-1}N) < 1$. □

Corollary 4 *Let $\mathscr{G}$ be connected, $\Sigma_{\mathbf{z}_1}, \ldots, \Sigma_{\mathbf{z}_m}$ be the covariance matrices associated to m independent p−dimensional measurements, and $\Sigma_{\mathbf{z}}$ be their associated block-diagonal covariance matrix as in Eq.* (3.33)*. Consider that each robot $i \in \mathscr{V}$ executes the Algorithm 3 to update its variable $\hat{\mathbf{x}}^a_i(t)$. Then, for all $i \in \mathscr{V}$,*

$$\lim_{t \to \infty} \hat{\mathbf{x}}^a_i(t) = \hat{\mathbf{x}}^a_i, \tag{3.44}$$

converges to the anchor-based centralized solution $\hat{\mathbf{x}}^a_i$ given by Eq. (3.34)*.* □

[1] $A \succ B$ ($A \succeq B$) represent that matrix $A - B$ is positive-definite (positive-semidefinite). Equivalently, $\prec$, $\preceq$ are used for negative-definite and negative-semidefinite matrices.

3.4.2 Centralized Centroid-Based Position Estimation

The accuracy of the estimated states $\hat{\mathbf{x}}^a_{\mathcal{V}}$, $\Sigma_{\hat{\mathbf{x}}^a_{\mathcal{V}}}$ in anchor-based approaches depend on the selected anchor a. Instead of that it is more interesting to compute the states of the robots $\hat{\mathbf{x}}^{cen}_{\mathcal{V}}$, $\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathcal{V}}}$ relative to the *centroid* given by the average of the states,

$$\hat{\mathbf{x}}^{cen}_{\mathcal{V}} = (\mathbf{I} - H_{cen})\,\hat{\mathbf{x}}^a_{\mathcal{V}}, \quad \Sigma_{\hat{\mathbf{x}}^{cen}_{\mathcal{V}}} = (\mathbf{I} - H_{cen})\,\Sigma_{\hat{\mathbf{x}}^a_{\mathcal{V}}}\,(\mathbf{I} - H_{cen})^T, \tag{3.45}$$
$$\text{where } H_{cen} = \left(\mathbf{1}_n \otimes \mathbf{I}_p\right)\left(\mathbf{1}_n \otimes \mathbf{I}_p\right)^T/n.$$

The value of this representation is that the states of the robots $\hat{\mathbf{x}}^{cen}_{\mathcal{V}}$, $\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathcal{V}}}$ with respect to the centroid are the same regardless of the anchor robot, i.e., the centroid solution is unique. Additionally, as the following result shows, it produces more accurate estimates than the ones provided by any anchor selection. We compare the block-traces[2] blkTr of their covariance matrices [6].

Proposition 5 *The covariance matrices of the centroid-based $\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathcal{V}}}$ and anchor-based $\Sigma_{\hat{\mathbf{x}}^a_{\mathcal{V}}}$ estimates satisfy, for all anchors $a \in \mathcal{V}$,*

$$\text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathcal{V}}}\right) \preceq \text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^a_{\mathcal{V}}}\right), \qquad \text{Tr}\left(\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathcal{V}}}\right) \leq \text{Tr}\left(\Sigma_{\hat{\mathbf{x}}^a_{\mathcal{V}}}\right). \tag{3.46}$$

Proof Let P_{ij} and Q_{ij} be the $p \times p$ blocks of, respectively, the anchor and the centroid-based covariances, $\Sigma_{\hat{\mathbf{x}}^a_{\mathcal{V}}} = [P_{ij}]$, $\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathcal{V}}} = [Q_{ij}]$ with $i, j \in \mathcal{V}$. The block-trace of the anchor-based covariance matrix is

$$\text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^a_{\mathcal{V}}}\right) = \sum_{i=1}^{n} P_{ii}. \tag{3.47}$$

Considering Eq. (3.45), each block in the main diagonal of the centroid-based $\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathcal{V}}}$ covariance matrix is given by

$$Q_{ii} = P_{ii} - \frac{1}{n}\sum_{j=1}^{n}\left(P_{ij} + P_{ji}\right) + \frac{1}{n^2}\sum_{j=1}^{n}\sum_{j'=1}^{n} P_{jj'}, \tag{3.48}$$

for $i \in \mathcal{V}$, and thus its block-trace is

$$\begin{aligned}\text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathcal{V}}}\right) &= \sum_{i=1}^{n} Q_{ii} = \sum_{i=1}^{n} P_{ii} - \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} P_{ij} \\ &= \text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^a_{\mathcal{V}}}\right) - (\mathbf{1}_n \otimes \mathbf{I}_p)^T\,\Sigma_{\hat{\mathbf{x}}^a_{\mathcal{V}}}\,(\mathbf{1}_n \otimes \mathbf{I}_p)/n.\end{aligned} \tag{3.49}$$

[2] The block-trace of a matrix defined by blocks $P = [P_{ij}]$ with $i, j \in \{1, \ldots, n\}$ is the sum of its diagonal blocks, $\text{blkTr}(P) = \sum_{i=1}^{n} P_{ii}$.

Since $\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}}$ is symmetric and positive-semidefinite, then $(\mathbf{1}_n \otimes \mathbf{I}_p)^T \Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}} (\mathbf{1}_n \otimes \mathbf{I}_p) \succeq \mathbf{0}$, and thus $\text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathscr{V}}}\right) - \text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}}\right) \preceq \mathbf{0}$, as in Eq. (3.46). Observe that the trace of the block-trace of a matrix A is equal to its trace, $\text{Tr}(\text{blkTr}(A)) = \text{Tr}(A)$. Since $\text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathscr{V}}}\right) - \text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}}\right) \preceq \mathbf{0}$, the elements in the main diagonal of $\text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathscr{V}}}\right)$ are smaller than or equal to the ones in the main diagonal of $\text{blkTr}\left(\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}}\right)$ so that

$$\text{Tr}(\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathscr{V}}}) = \text{Tr}(\text{blkTr}(\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathscr{V}}})) \leq \text{Tr}(\text{blkTr}(\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}})) = \text{Tr}(\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}}).$$

□

In particular, from Eq. (3.49), $\text{Tr}(\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}}) - \text{Tr}(\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathscr{V}}}) = \frac{1}{n}\sum_{i=1}^n \sum_{j=1}^n \text{Tr}(P_{ij})$. Note that the previous result holds when the anchor state $\hat{\mathbf{x}}^a_a$ is set to a general value, not necessarily $\mathbf{0}$. It also holds when there is more than one anchor. Consider that the first k robots are anchors. In this case, matrix $\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}} = [P_{ij}]$ has its blocks $P_{ij} = \mathbf{0}$ for $i, j \in \{1, \ldots, k\}$, and Eq. (3.49) gives $\text{blkTr}(\Sigma_{\hat{\mathbf{x}}^{cen}_{\mathscr{V}}}) = \text{blkTr}(\Sigma_{\hat{\mathbf{x}}^a_{\mathscr{V}}}) - \sum_{i=k+1}^n \sum_{j=k+1}^n P_{ij}/n$, where $\sum_{i=k+1}^n \sum_{j=k+1}^n P_{ij}/n \succeq \mathbf{0}$.

We propose an algorithm that allows each robot $i \in \mathscr{V}$ to compute its state $\hat{\mathbf{x}}^{cen}_i$ with respect to the centroid in a distributed fashion, where $\hat{\mathbf{x}}^{cen}_{\mathscr{V}}$ is given in Eq. (3.45), $\hat{\mathbf{x}}^{cen}_{\mathscr{V}} = ((\hat{\mathbf{x}}^{cen}_1)^T, \ldots, (\hat{\mathbf{x}}^{cen}_n)^T)^T$. These states sum up to zero, $\hat{\mathbf{x}}^{cen}_1 + \cdots + \hat{\mathbf{x}}^{cen}_n = \mathbf{0}$, since $(\mathbf{1}_n \otimes \mathbf{I}_p)(\mathbf{I} - H_{cen}) = \mathbf{0}$, and for neighboring robots i and j satisfy $\hat{\mathbf{x}}^{cen}_i = \hat{\mathbf{x}}^{cen}_j - \hat{\mathbf{x}}^a_j + \hat{\mathbf{x}}^a_i$. Thus, a straightforward solution would consist of firstly computing the anchor-based states of the robots $\hat{\mathbf{x}}^a_{\mathscr{V}} = ((\hat{\mathbf{x}}^a_1)^T, \ldots, (\hat{\mathbf{x}}^a_n)^T)^T$, and in a second phase initializing the robots' variables so that they sum up to zero, $\hat{\mathbf{x}}^{cen}_i(0) = \mathbf{0}$, for $i \in \mathscr{V}$, and updating them at each step t with an averaging algorithm that conserves the sum:

$$\hat{\mathbf{x}}^{cen}_i(t+1) = \sum_{j \in \mathscr{N}_i \cup \{i\}} \mathscr{W}_{i,j}(\hat{\mathbf{x}}^{cen}_j(t) - \hat{\mathbf{x}}^a_j + \hat{\mathbf{x}}^a_i), \tag{3.50}$$

for $i \in \mathscr{V}$, where $\mathscr{W} = [\mathscr{W}_{i,j}]$ is a doubly stochastic weight matrix such that $\mathscr{W}_{i,j} > 0$ if $(i, j) \in \mathscr{E}$ and $\mathscr{W}_{i,j} = 0$ when $j \notin \mathscr{N}_i$. Besides, $\mathscr{W}_{i,i} \in [\alpha, 1]$, $\mathscr{W}_{i,j} \in \{0\} \cup [\alpha, 1]$ for all $i, j \in \mathscr{V}$, for some $\alpha \in (0, 1]$. More information about averaging algorithms can be found in Appendix A and at [9, 26, 37]. The term $-\hat{\mathbf{x}}^a_j + \hat{\mathbf{x}}^a_i$ is the relative measurement $\mathbf{z}_e$ with $e = (j, i)$ for noise free scenarios, and the optimal or corrected measurement [28] $\hat{\mathbf{z}}_e$ for the noisy case, $\hat{\mathbf{z}} = (\mathscr{A} \otimes \mathbf{I}_p)^T \hat{\mathbf{x}}^a_{\mathscr{V}}$, with $\hat{\mathbf{z}} = ((\hat{\mathbf{z}}_1)^T, \ldots, (\hat{\mathbf{z}}_m)^T)^T$. In what follows we propose an algorithm where, at each iteration t, (3.50) is executed not on the exact $\hat{\mathbf{x}}^a_i$, $\hat{\mathbf{x}}^a_j$, but on the most recent estimates $\hat{\mathbf{x}}^a_i(t)$, $\hat{\mathbf{x}}^a_j(t)$ obtained with Algorithm 3.

3.4.3 Distributed Centroid-Based Position Estimation

Now we study a distributed localization algorithm for estimating the position the robots relative to the centroid.

Algorithm 4 Let each robot $i \in \mathscr{V}$ have an estimate of its own state relative to the centroid, $\hat{\mathbf{x}}_i^{cen}(t) \in \mathbb{R}^p$, initialized at $t = 0$ with $\hat{\mathbf{x}}_i^{cen}(0) = \mathbf{0}$. At each time step t, each robot $i \in \mathscr{V}$ updates $\hat{\mathbf{x}}_i^{cen}(t)$ with

$$\hat{\mathbf{x}}_i^{cen}(t+1) = \sum_{j \in \mathscr{N}_i \cup \{i\}} \mathscr{W}_{i,j} (\hat{\mathbf{x}}_j^{cen}(t) + \hat{\mathbf{x}}_i^{a}(t) - \hat{\mathbf{x}}_j^{a}(t)), \tag{3.51}$$

where $\hat{\mathbf{x}}_i^{a}(t)$, $\hat{\mathbf{x}}_j^{a}(t)$ are the most recent estimates that robots i and j have at iteration t of the variables in Algorithm 3 and $\mathscr{W}_{i,j}$ are the Metropolis weights as defined in Eq. (A.3) in Appendix A.

Theorem 7 *Let all the robots $i \in \mathscr{V}$ execute the Algorithm 4 and let $\mathscr{G}$ be connected. Then, the estimated states $\hat{\mathbf{x}}_i^{cen}(t)$ at each robot $i \in \mathscr{V}$ asymptotically converge to the state of i relative to the centroid $\hat{\mathbf{x}}_i^{cen}$ given by Eq. (3.45),*

$$\lim_{t \to \infty} \hat{\mathbf{x}}_i^{cen}(t) = \hat{\mathbf{x}}_i^{cen}. \tag{3.52}$$

Let $\mathbf{e}_{cen}(t) = \left[(\hat{\mathbf{x}}_1^{cen}(t) - \hat{\mathbf{x}}_1^{cen})^T, \ldots, (\hat{\mathbf{x}}_n^{cen}(t) - \hat{\mathbf{x}}_n^{cen})^T\right]^T$ be the error vector containing the estimation errors of the n robots at iteration t. For fixed communication graphs $\mathscr{G}$, the norm of the error vector after t iterations of Algorithm 4 satisfies

$$||\mathbf{e}_{cen}(t)||_2 \leq \lambda_{\text{eff}}^t(\mathscr{W})||\mathbf{e}_{cen}(0)||_2 + 2p(n-1)\sigma_J \lambda_{\text{eff}}^t(\mathscr{W}) \sum_{k=1}^{t} \left(\frac{\rho(J)}{\lambda_{\text{eff}}(\mathscr{W})}\right)^k, \tag{3.53}$$

where J is the Jacobi matrix $J = D^{-1}N$, with D and N computed as in Eqs. (3.35)–(3.37), σ_J is a constant that depends on the initial Jacobi error and on J. $\mathscr{W}$ is the Metropolis weight matrix as defined in Eq. (A.3) in Appendix A, and $\mathbf{e}_{cen}(0)$ is the initial error at $t = 0$.

Proof First of all, we derive the expression for the convergence rate in Eq. (3.53). We express Algorithm 4 in terms of the error vectors associated to the centroid $\mathbf{e}_{cen}(t)$ and the anchor-based $\mathbf{e}_a(t) \in \mathbb{R}^{(n-1)p}$ estimation methods (Algorithms 3 and 4),

$$\mathbf{e}_{cen}(t) = \left[(\hat{\mathbf{x}}_1^{cen}(t))^T, \ldots, (\hat{\mathbf{x}}_n^{cen}(t))^T\right]^T - \hat{\mathbf{x}}_{\mathscr{V}}^{cen},$$

with $\hat{\mathbf{x}}^{cen}_{\mathcal{V}} = \left[(\hat{\mathbf{x}}^{cen}_1)^T, \ldots, (\hat{\mathbf{x}}^{cen}_n)^T\right]^T$ given by Eq. (3.45), and

$$\tilde{\mathbf{e}}_a(t) = \left[(\hat{\mathbf{x}}^a_2(t)^T, \ldots, \hat{\mathbf{x}}^a_n(t)^T\right]^T - \hat{\mathbf{x}}^a_{\mathcal{V}^a},$$

with $\hat{\mathbf{x}}^a_{\mathcal{V}^a} = \left[(\hat{\mathbf{x}}^a_2)^T, \ldots, (\hat{\mathbf{x}}^a_n)^T\right]^T$ given by Eq. (3.34), where for simplicity we let the robot $i = 1$ be the anchor a. We let $\mathbf{e}_a(t)$ be $(\mathbf{0}_p^T, \tilde{\mathbf{e}}_a(t)^T)^T$. Recall that $\sum_{j \in \mathcal{N}_i \cup \{i\}} \hat{\mathbf{x}}^a_i(t) = \hat{\mathbf{x}}^a_i(t)$ and that the estimated states relative to the centroid $\hat{\mathbf{x}}^{cen}_{\mathcal{V}}$ are $\hat{\mathbf{x}}^{cen}_{\mathcal{V}} = (\mathbf{I} - H_{cen})\hat{\mathbf{x}}^a_{\mathcal{V}}$ as in Eq. (3.45). Algorithm 4 becomes

$$\mathbf{e}_{cen}(t) = (\mathscr{W} \otimes \mathbf{I}_p)\mathbf{e}_{cen}(t-1) + ((\mathbf{I}_n - \mathscr{W}) \otimes \mathbf{I}_p)\mathbf{e}_a(t-1) + P\hat{\mathbf{x}}^a_{\mathcal{V}}, \tag{3.54}$$

where the term P that is multiplying $\hat{\mathbf{x}}^a_{\mathcal{V}}$ is

$$P = \mathbf{I} - (\mathscr{W} \otimes \mathbf{I}_p) - (\mathbf{I} - (\mathscr{W} \otimes \mathbf{I}_p))(\mathbf{I} - H_{cen}) = (\mathbf{I} - (\mathscr{W} \otimes \mathbf{I}_p))H_{cen}. \tag{3.55}$$

We use the fact that $(\mathscr{W} \otimes \mathbf{I}_p)H_{cen} = H_{cen}$, and the previous expression gives $P = \mathbf{0}$ and Eq. (3.54) becomes

$$\begin{aligned}\mathbf{e}_{cen}(t) &= (\mathscr{W} \otimes \mathbf{I}_p)\mathbf{e}_{cen}(t-1) + ((\mathbf{I}_n - \mathscr{W}) \otimes \mathbf{I}_p)\mathbf{e}_a(t-1)\\ &= (\mathscr{W} \otimes \mathbf{I}_p)^t \mathbf{e}_{cen}(0) + \sum_{k=0}^{t-1} (\mathscr{W} \otimes \mathbf{I}_p)^{t-k-1}\left((\mathbf{I} - \mathscr{W}) \otimes \mathbf{I}_p\right)\mathbf{e}_a(k).\end{aligned} \tag{3.56}$$

Then, the norm of the error $\mathbf{e}_{cen}(t)$ satisfies

$$\|\mathbf{e}_{cen}(t)\|_2 \leq \lambda^t_{\text{eff}}(\mathscr{W})\|\mathbf{e}_{cen}(0)\|_2 + 2\sum_{k=0}^{t-1} \lambda^{t-k-1}_{\text{eff}}(\mathscr{W})\|\mathbf{e}_a(k)\|_2, \tag{3.57}$$

where we have used the fact that $\|\left((\mathscr{W} - \mathbf{I}) \otimes \mathbf{I}_p\right)\|_2 \leq 2$ since $\mathscr{W}$ is the Metropolis weight matrix given by Eq. (A.3) in Appendix A.

We analyze now the norm of error $\|\mathbf{e}_a(t)\|_2$, which is related to the error vector of the Jacobi algorithm $\tilde{\mathbf{e}}_a(t) \in \mathbb{R}^{(n-1)p}$ by $\mathbf{e}_a(t) = (\mathbf{0}, \tilde{\mathbf{e}}^T_a(t))^T$. Let J be the Jacobi matrix, and $V_J = \left[\mathbf{v}_{p+1}(J), \ldots, \mathbf{v}_{np}(J)\right]$ and $\lambda_J = \text{Diag}\left(\lambda_{p+1}(J), \ldots, \lambda_{np}(J)\right)$ be its associated eigenvectors and eigenvalues so that $J = V_J\, \lambda_J\, V_J^{-1}$, and $||\mathbf{v}_i(J)||_2 = 1$. The error vector $\tilde{\mathbf{e}}_a(t)$ evolves according to

$$\tilde{\mathbf{e}}_a(t) = J\tilde{\mathbf{e}}_a(t-1) = J^t\tilde{\mathbf{e}}_a(0). \tag{3.58}$$

For each initial error vector $\tilde{\mathbf{e}}_a(0)$ there exist $\sigma_{p+1}, \ldots, \sigma_{np}$ such that

$$\tilde{\mathbf{e}}_a(0) = \sum_{i=p+1}^{np} \sigma_i \mathbf{v}_i(J),$$

and then the error vector $\tilde{\mathbf{e}}_a(t)$ after t iterations of the Jacobi algorithm given by Eq. (3.58) can be expressed as

$$\tilde{\mathbf{e}}_a(t) = V_J \lambda_J^t V_J^{-1} V_J \left[\sigma_{p+1}, \ldots, \sigma_{np}\right]^T = \sum_{i=p+1}^{np} \sigma_i \mathbf{v}_i(J) \lambda_i^t(J).$$

Let $\sigma_J = \max_{i=p+1}^{np} |\sigma_i|$, and $\rho(J) = \max_{i=p+1}^{np} |\lambda_i(J)|$. For all $t \geq 0$, the norm of the error vector $||\tilde{\mathbf{e}}_a(t)||_2$ satisfies

$$||\mathbf{e}_a(t)||_2 = ||\tilde{\mathbf{e}}_a(t)||_2 \leq p(n-1)\sigma_J \rho^t(J). \tag{3.59}$$

Linking this with Eq. (3.57) gives that the convergence rate is

$$\|\mathbf{e}_{cen}(t)\|_2 \leq \lambda_{\text{eff}}^t(\mathscr{W})\|\mathbf{e}_{cen}(0)\|_2 + 2p(n-1)\sigma_J \sum_{k=0}^{t-1} \lambda_{\text{eff}}^{t-k-1}(\mathscr{W})\rho^k(J), \tag{3.60}$$

as in Eq. (3.53).

Now we prove the asymptotical convergence to the centroid (3.52). If both the Jacobi and the general algorithm have the same convergence rate, $\rho(J) = \lambda_{\text{eff}}(\mathscr{W})$, then Eq. (3.60) gives

$$||\mathbf{e}_{cen}(t)||_2 \leq \lambda_{\text{eff}}^t(\mathscr{W})||\mathbf{e}_{cen}(0)||_2 + 2p(n-1)\sigma_J \lambda_{\text{eff}}^{t-1}(\mathscr{W})t, \tag{3.61}$$

whereas for $\rho(J) \neq \lambda_{\text{eff}}(\mathscr{W})$, it gives

$$||\mathbf{e}_{cen}(t)||_2 \leq \lambda_{\text{eff}}^t(\mathscr{W})||\mathbf{e}_{cen}(0)||_2 + \frac{2p(n-1)\sigma_J}{\rho(J) - \lambda_{\text{eff}}(\mathscr{W})}(\rho^t(J) - \lambda_{\text{eff}}^t(\mathscr{W})). \tag{3.62}$$

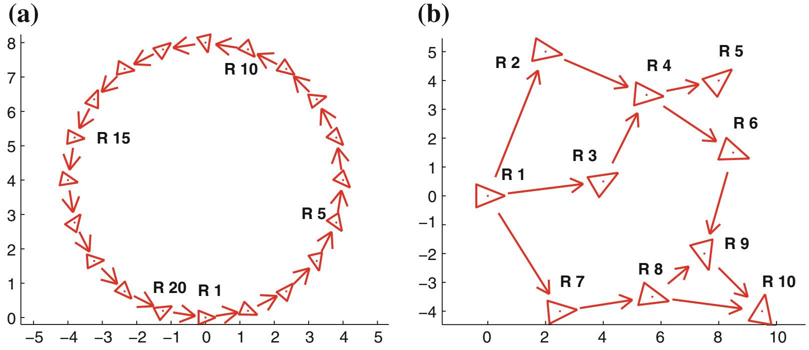
Note that $\lambda_{\text{eff}}(\mathscr{W}) < 1$ for connected graphs $\mathscr{G}$. Then, the term $\lambda_{\text{eff}}^t(\mathscr{W})||\mathbf{e}_{cen}(0)||_2$ in Eqs. (3.61) and (3.62) exponentially tends to zero as $t \to \infty$ regardless of the initial error $\mathbf{e}_{cen}(0)$. For the case $\rho(J) = \lambda_{\text{eff}}(\mathscr{W})$, the term $\lambda_{\text{eff}}^t(\mathscr{W})t$ in Eq. (3.61) is decreasing for $t \geq \frac{\lambda_{\text{eff}}(\mathscr{W})}{1-\lambda_{\text{eff}}(\mathscr{W})}$ and thus it tends to zero as $t \to \infty$. For $\rho(J) \neq \lambda_{\text{eff}}(\mathscr{W})$, the term $(\rho^t(J) - \lambda_{\text{eff}}^t(\mathscr{W}))$ in Eq. (3.62) asymptotically tends to zero since $\lambda_{\text{eff}}(\mathscr{W})$ is less than 1, and as stated by Theorem 6, $\rho(J) < 1$. Therefore, $\lim_{t\to\infty} ||\mathbf{e}_{cen}(t)||_2 = 0$, where $||\mathbf{e}_{cen}(t)||_2 = 0$ iff $\mathbf{e}_{cen}(t) = 0$, what concludes the proof. □

3.5 Simulations

Planar Localization from Noisy Measurements

A set of simulations have been carried out to show the performance of the method for planar localization from noisy measurements, and to compare the results of the centralized (Sect. 3.3.1) and the distributed (Sect. 3.3.2) approaches.

First, a team of $n = 20$ robots are placed in a ring of radius 4 m with their orientations randomly generated within $\pm\frac{\pi}{2}$ (Fig. 3.1a). Each robot measures the relative pose of the next robot, with noises in the x- and y- coordinates of 6 cm standard deviation, and of 1 degree for the orientation. The robots execute the proposed method to compute their pose with respect to the anchor node $R1$. The experiment is repeated 100 times and the average results can be seen in Table 3.1. The first rows show the solution computed by the localization algorithm in Sect. 3.3.1, and the next rows compare the distributed implementation of the algorithm (Sect. 3.3.2) against the results obtained by the centralized algorithm in Sect. 3.3.1. We use the flagged initialization [4] that is known to produce fast convergence results. The convergence speeds during the first and the third phases depend on the values of respectively $\rho(C^{-1}D)$ in (3.18) and $\rho(\mathbf{I} - M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}^a}})$ in (3.31), which here are close to one (slow convergence). The second phase is always executed in a single iteration (it does not have any convergence speed associated). After executing the first phase for $t = 50$ iterations, the obtained $\bar{\theta}^a_{\mathcal{V}}$ still differs from the centralized solution $\tilde{\theta}^a_{\mathcal{V}}$ by around $0.16°$. If we increase the number of iterations we obtain better approximations that differ only by 0.01 ($t = 100$) and $8.5e - 05$ ($t = 200$) degrees. The next three rows show the results after executing the second phase followed by 200 iterations of the third phase. Since the second and third phases have been executed using $\bar{\theta}^a_{\mathcal{V}}$ instead of $\tilde{\theta}^a_{\mathcal{V}}$, the final results also differ. For the case $t = 200$ (third column), the difference between the pose estimated by the distributed and centralized approaches



20 robots in a ring of 4m radius

10 robots in general positions

Fig. 3.1 Scenarios tested. Each robot (*triangles*) measures the relative pose of other team members (*outgoing arrows*). Robots connected by an arrow can exchange data

Table 3.1 Results for the scenario in Fig. 3.1a

Localization results versus ground truth			
	Max error	Average standard deviation	
Orientation phase 1	3.38°	1.87°	
x-coordinate phase 3	27.85 cm	13.45 cm	
y-coordinate phase 3	24.33 cm	12.31 cm	
Orientation phase 3	4.03°	1.66°	
Distributed implementation (flagged-initialization)			
$\rho(C^{-1}D)$	0.99		
$\rho(\mathbf{I} - M^{-1}\Upsilon_{\hat{\mathbf{q}}^a_{\mathcal{V}a}})$	0.99		
Max error	$t = 50$	$t = 100$	$t = 200$
Orientation phase 1	0.16°	0.01°	$8.5e - 05$°
x-coordinate phase 3	1.74 cm	1.64 cm	1.64 cm
y-coordinate phase 3	0.84 cm	0.49 cm	0.48 cm
Orientation phase 3	0.29°	0.12°	0.11°

is small (1.64 and 0.48 cm for the x- and y- coordinates, and 0.11 degrees for the orientation), and similar results are obtained for $t = 100$.

Other simulation with 10 robots placed as in Fig. 3.1b has been carried out. If there is an arrow from robot i into j, then robot i measures the relative pose of robot j, with additive noises of 2.5 degrees of standard deviation for the orientation, and of 5 % d and 0.7 % d standard deviation for respectively the x and y-coordinates, where d is the distance between the robots. The robots execute the distributed algorithm during the phase 1 to compute their orientations with respect to the anchor node R1 (Fig. 3.2a), obtaining estimates (blue) very close to the ground truth data (red). They execute phase 2 to express the relative position measurements in the reference

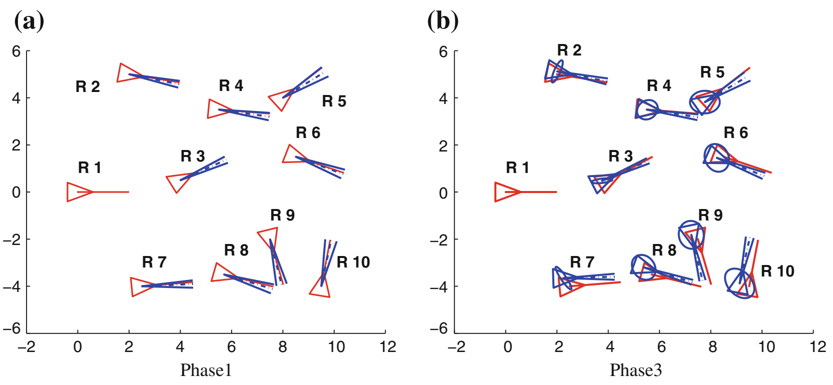


Fig. 3.2 The robots estimate their poses (*blue dashed*) relative to the anchor R1 for the experiment in Fig. 3.1b. The ground truth data (*red solid*) and the covariances computed by the centralized approach (*blue solid*) are also displayed

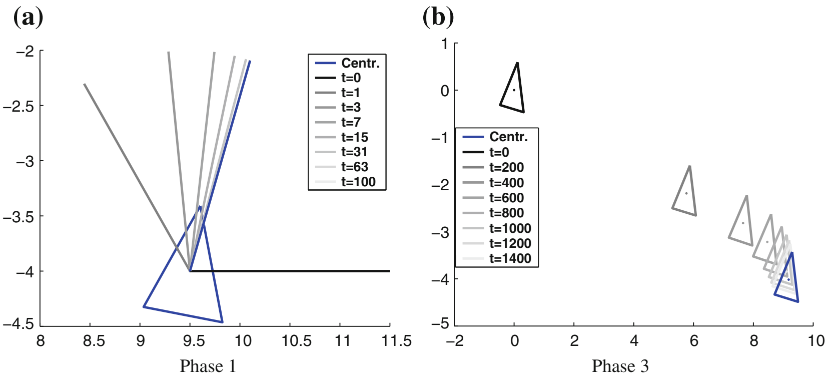


Fig. 3.3 Detail of phases 1 and 3 of the proposed strategy. The estimates of robot R10 (*gray*) successively approach the centralized solution (*blue*)

frame of the anchor node. Finally, they execute the phase 3 to obtain both, their positions and orientations relative to the anchor node (Fig. 3.2b). Figure 3.3 shows a detail of the iterations during phases 1 and 3. Although the convergence was previously proved only for graphs with low connectivity (ring or string graphs), in the experiments with general communication graphs the algorithm has been found to converge as well.

Centroid-Based Noisy Position Localization

We study the performance of the algorithm presented in Sect. 3.4 in a multi-robot localization scenario (Fig. 3.4) with $n = 20$ robots (black circles) that get noisy measurements (gray crosses and ellipses) of the position of robots which are closer than

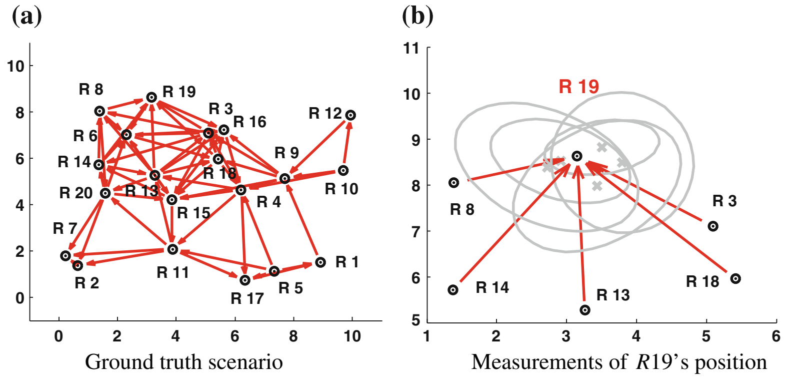


Fig. 3.4 **a** 20 robots (*black circles*) are placed randomly in a region of 10 × 10 m. There is an edge $e = (i, j) \in \mathscr{E}$ (*red arrows*) between pairs of robots that are closer than 4 m. **b** Each robot i has a noisy measurement $\mathbf{z}_e$ (*gray crosses* and *ellipses*) of the relative position of its out-neighbors j, with $e = (i, j)$. The noises are proportional to the distance between the robots

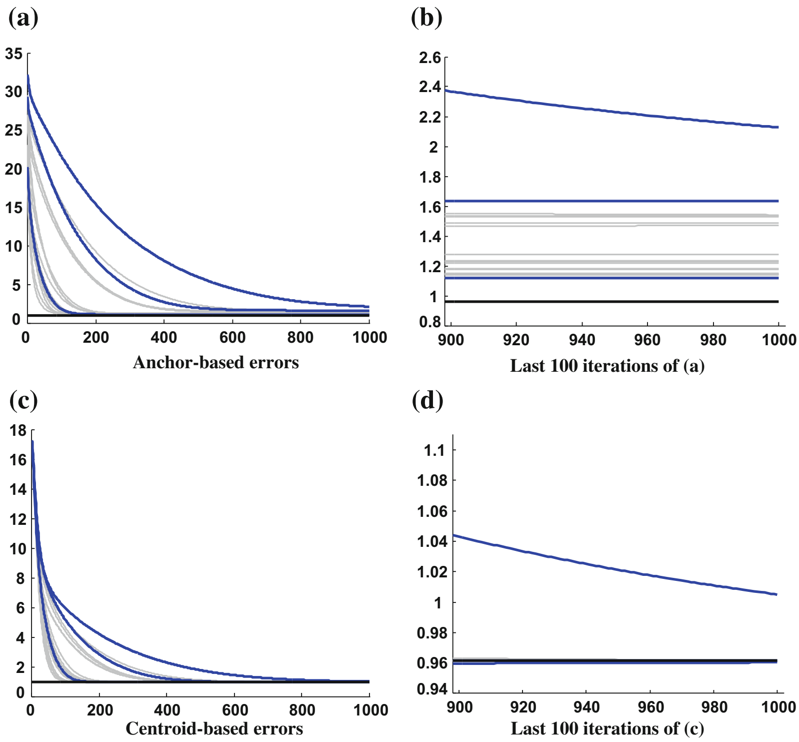


Fig. 3.5 The experiment in Fig. 3.4 is generated 100 times with the same noise levels but different noise values. We display the average norm of the error with the difference between the estimates and the ground truth for the 100 different experiments. **a** Results of Algorithm 3 when each robot $i \in \mathcal{V}$ is used as the anchor (*gray lines*). The special cases that the anchor is $R1$, $R3$ and $R12$ are shown in *blue*. The *black line* is the asymptotic error reached with the centroid-based estimation. **b** Detail of iterations 900–1000 in (*a*). **c** Results of Algorithm 4 using all the possible anchors. **d** Detail of iterations 900–1000 in (*c*)

4 m. We analyze the states estimated by the n robots along 1000 iterations of the proposed algorithm (Fig. 3.5). Robots initialize their states $\hat{\mathbf{x}}_i^a(t)$, $\hat{\mathbf{x}}_i^{cen}(t)$ with zeros and execute Algorithms 3 and 4. We generate specific noises as the ones in Fig. 3.4 for 100 different samples. For each of them, we record the norm of the error vector containing the difference between the estimates at the n robots and the ground-truth positions at each iteration t. In Fig. 3.5a we show the results of Algorithm 3 when each robot $i \in \mathcal{V}$ is used as the anchor (gray lines). The most and least precise anchor-based results, which are obtained for respectively $R3$ and $R12$, are shown in blue. The results for robot $R1$, which is conventionally used as the anchor, are displayed in blue as well. The black line is the asymptotic error reached with the centroid-based estimation method. As it can be seen, the errors reached with the anchor-based solutions are greater than the ones associated to the centroid. This is even more evident in Fig. 3.5b, which shows the last 100 iterations in Fig. 3.5a. In

Fig. 3.5c we show the equivalent errors for the centroid-based estimation algorithm (Algorithm 4), using all the possible anchors for Algorithm 3. Here, in all cases the error estimates (gray lines) converge to the asymptotic error of the centroid-based estimation method (black line).

3.6 Closure

Along this chapter, the problem of network localization has been studied for different scenarios: the estimation of the planar localization with respect to an anchor from noisy relative measurements, and the estimation of higher dimension positions with respect to the, simultaneously computed, centroid of the network using also noisy measurements. We have analyzed distributed strategies that allow the robots to agree on a common global frame, and to compute their poses relative to the global frame. The presented algorithms exclusively rely on local computations and data exchange with direct neighbors. Besides, they only require the robots to maintain their own estimated poses relative to the common frame. Thus, the memory load of the algorithm is low compared to methods where each robot must also estimate the positions or poses of any other robot. We have discussed the performance of the planar pose localization algorithm relative to an anchor node, for ring or string topologies. The centroid-based position localization method has been studied to produce more accurate results than any anchor-based solution. Besides, in the experiments we have shown that it converges faster than the anchor-based solutions.

References

1. B.D.O. Anderson, I. Shames, G. Mao, B. Fidan, Formal theory of noisy sensor network localization. SIAM J. Discret. Math. **24**(2), 684–698 (2010)
2. R. Aragues, L. Carlone, G. Calafiore, C. Sagues, Multi agent localization from noisy relative pose measurements, in *IEEE International Conference on Robotics and Automation* , Shanghai, China, May 2011, pp. 364–369
3. R. Aragues, L. Carlone, C. Sagues, G. Calafiore, Distributed centroid estimation from noisy relative measurements. Syst. Control Lett. **61**(1), 773–779 (2012)
4. P. Barooah, J. Hespanha, Distributed estimation from relative measurements in sensor networks, in *International Conference on Intelligent Sensing and Information Processing*, Chennai, India, January 2005, pp. 88–93
5. P. Barooah, J. Hespanha, Estimation on graphs from relative measurements. IEEE Control Syst. Mag. **27**(4), 57–74 (2007)
6. P. Barooah, J. Hespanha, Error scaling laws for linear optimal estimation from relative measurements. IEEE Trans. Inf. Theory **55**(12), 5661–5673 (2009)
7. D.J. Bennet, C.R. McInnes, Distributed control of multi-robot systems using bifurcating potential fields. Robot. Auton. Syst. **58**(3), 256–264 (2010)
8. D.P. Bertsekas, J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods* (Athena Scientific, 1997)

9. F. Bullo, J. Cortes, S. Martinez, *Distributed Control of Robotic Networks*. Applied Mathematics Series (Princeton University Press, 2009). Electronically available at http://coordinationbook.info
10. G. Calafiore, L. Carlone, M. Wei, A distributed gauss-newton approach for range-based localization of multi agent formations, in *IEEE Multi-Conference on Systems and Control*, Yokohama, Japan, September 2010, pp. 1152–1157
11. G. Calafiore, L. Carlone, M. Wei, A distributed gradient method for localization of formations using relative range measurements, in *IEEE Multi-Conference on Systems and Control*, Yokohama, Japan, September 2010, pp. 1146–1151
12. L. Carlone, R. Aragues, J.A. Castellanos, B. Bona, A first-order solution to simultaneous localization and mapping with graphical models, in *IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 1764–1771
13. L. Carlone, R. Aragues, J.A. Castellanos, B. Bona, A linear approximation for graphbased simultaneous localization and mapping, in *Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011
14. S. Carpin, Fast and accurate map merging for multi-robot systems. Auton. Robot. **25**(3), 305–316 (2008)
15. S. Carpin, A. Birk, V. Jucikas, On map merging. Robot. Auton. Syst. **53**(1), 1–14 (2005)
16. J. Cortes, Global and robust formation-shape stabilization of relative sensing networks. Automatica **45**(12), 2754–2762 (2009)
17. L. Elsner, V. Mehrmann, Convergence of block iterative methods for linear systems arising in the numerical solution of Euler equations. Numerische Mathematik **59**(1), 541–559 (1991)
18. J.A. Fax, R.M. Murray, Information flow and cooperative control of vehicle formations. IEEE Trans. Autom. Control **49**(9), 1465–1476 (2004)
19. M. Franceschelli, A. Gasparri, On agreement problems with gossip algorithms in absence of common reference frames, in *IEEE International Conference on Robotics and Automation*, Anchorage, USA, May 2010, pp. 4481–4486
20. J.J. Guerrero, A.C. Murillo, C. Sagues, Localization and matching using the planar trifocal tensor with bearing-only data. IEEE Trans. Robot. **24**(2), 494–501 (2008)
21. M. Ji, M. Egerstedt, Distributed coordination control of multiagent systems while preserving connectedness. IEEE Trans. Robot. **23**(4), 693–703 (2007)
22. J. Knuth, P. Barooah, Distributed collaborative localization of multiple vehicles from relative pose measurements, in *Allerton Conference on Communications, Control and Computing*, Urbana-Champaign, USA, October 2009, pp. 314–321
23. L. Kolotilina, Bounds for eigenvalues of symmetric block Jacobi scaled matrices. J. Math. Sci. **79**(3), 1043–1047 (1996)
24. G. Lafferriere, A. Williams, J. Caughman, J.J.P. Veerman, Decentralized control of vehicle formations. Syst. Control Lett. **54**(9), 899–910 (2005)
25. N. Mostagh, A. Jadbabaie, Distributed geodesic control laws for flocking of nonholonomic agents. IEEE Trans. Autom. Control **52**(4), 681–686 (2007)
26. W. Ren, R.W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control Communications and Control Engineering* (Springer, London, 2008)
27. S.I. Roumeliotis, G.A. Bekey, Distributed multirobot localization. IEEE Trans. Robot. Autom. **18**(5), 781–795 (2002)
28. W.J. Russell, D. Klein, J.P. Hespanha, Optimal estimation on the graph cycle space, in *American Control Conference*, Baltimore, June 2010, pp. 1918–1924
29. C. Sagues, A.C. Murillo, J.J. Guerrero, T. Goedemé, T. Tuytelaars, L. Van Gool, Localization with omnidirectional images using the 1D radial trifocal tensor, in *IEEE International Conference on Robotics and Automation*, Orlando, May 2006, pp. 551–556
30. A. Sarlette, R. Sepulchre, N.E. Leonard, Autonomous rigid body attitude synchronization. Automatica **45**(2), 572–577 (2008)
31. A. Savvides, W.L. Garber, R.L. Moses, M.B. Srivastava, An analysis of error inducing parameters in multihop sensor node localization. IEEE Trans. Mob. Comput. **4**(6), 567–577 (2005)

32. I. Skrjanc, G. Klancar, Optimal cooperative collision avoidance between multiple robots based on Bernstein-Bzier curves. Robot. Auton. Syst. **58**(1), 1–9 (2010)
33. S. Thrun, Y. Liu, Multi-robot SLAM with sparse extended information filters, in *International Symposium of Robotics Research*, Sienna, Italy, October 2003, pp. 254–266
34. N. Trawny, S.I. Roumeliotis, G.B. Giannakis, Cooperative multi-robot localization under communication constraints, in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, pp. 4394–4400
35. N. Trawny, X.S. Zhou, K.X. Zhou, S.I. Roumeliotis, Inter-robot transformations in 3-d. IEEE Trans. Robot. **26**(2), 226–243 (2010)
36. B. Varghese, G. McKee, A mathematical model, implementation and study of a swarm system. Robot. Auton. Syst. **58**(3), 287–294 (2010)
37. L. Xiao, S. Boyd, S. Lall, A scheme for robust distributed sensor fusion based on average consensus, in *Symposium on Information Processing of Sensor Networks (IPSN)*, Los Angeles, CA, April 2005, pp. 63–70
38. X.S. Zhou, S.I. Roumeliotis, Robot-to-robot relative pose estimation from range measurements. IEEE Trans. Robot. **24**(6), 1379–1393 (2008)