2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Multi-Agent Localization from Noisy Relative Pose Measurements

R. Aragues L. Carlone G. Calafiore C. Sagues

***Abstract*— In this paper we address the problem of estimating the poses of a team of agents when they do not share any common reference frame. Each agent is capable of measuring the relative position and orientation of its neighboring agents, however these measurements are not exact but they are corrupted with noises. The goal is to compute the pose of each agent relative to an anchor node. We present a strategy where, first of all, the agents compute their orientations relative to the anchor. After that, they update the relative position measurements according to these orientations, to finally compute their positions. As contribution we discuss the proposed strategy, that has the interesting property that can be executed in a distributed fashion. The distributed implementation allows each agent to recover its pose using exclusively local information and local interactions with its neighbors. This algorithm has a low memory load, since it only requires each node to maintain an estimate of its own orientation and position.**

I. INTRODUCTION

Multi-agent systems systems provide a natural robustness to individual failures and they are capable of accomplishing a task in a more efficient way than systems with a single agent. Many multi-agent tasks require the knowledge of the agent positions in some reference frame. However, typically the agents start at unknown poses and they do not share any common reference frame. In this paper we address the multi-agent localization problem, which consists of computing a set of agents poses so that they satisfy a set of noisy relative pose measurements.

The network localization problem has received a lot of attention lately. Distributed strategies have been presented [1], [2] that, based on distance measurements, compute the agents positions but do not retrieve their orientations. There exist many distributed algorithms that compute both the positions and orientations [3], [4] but assuming that the relative pose measurements are noise free. A similar problem is the formation control in relative sensing networks. The objective formation is defined by a set of restrictions on the relative poses between agents and the goal is that the agents move to positions satisfying these restrictions [5]–[7]. Although some works [5] study the effects of the noises in the algorithm performance, formation control methods are usually designed for noise free scenarios. However, relative measurements are typically corrupted with noises since they have been obtained from sensorial data. The noisy nature of the relative pose measurements has already been taken into account in the field of cooperative localization [8], [9], where a robotic team moves through an environment while estimating their poses. Each robot measures its own motion, and also the relative poses of nearby robots when available. Cooperative localization approaches, however, assume that an initial guess on the robot poses exists.

R. Aragues and C. Sagues are with the Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain raragues@unizar.es, csagues@unizar.es

L. Carlone and G. Calafiore are with CSPP, Laboratorio di Meccatronica, and with the Dipartimento di Automatica e Informatica, Politecnico di Torino, Italy luca.carlone@polito.it, giuseppe.calafiore@polito.it

Here instead, we assume that there is no initial knowledge of the agent poses. The only information available to each agent are the noisy relative poses of nearby nodes. There exist many methods that retrieve this information, see e.g., [10] and [11] for planar poses extracted from respectively distance and bearing-only measurements, or [12] for 3D scenarios and distance and bearing, bearing-only, and distance-only sensors. We assume that one of these methods is executed by the agents to compute the relative pose measurements and their associated covariances. We assume that the measurements are independent since they are acquired individually by the agents. The relative poses must be properly combined in order to produce an estimate of the network localization. Along this paper we present our network localization strategy that allows the use of distributed linear estimation algorithms [13]–[15].

This paper is organized as follows. Section II states the localization problem. Section III presents the proposed localization strategy for a centralized system. Section IV gives a distributed implementation of this strategy together with the cases where it can be applied. Section V evaluates the performance of the proposal under different simulated scenarios. Finally, Section VI presents the conclusions.

NOTATION

Throughout the paper we let $\mathbf{I}_r$ be the $r \times r$ identity matrix and $\mathbf{0}_{r\times r'}$ a $r \times r'$ matrix with all its elements equal to zero. The dimensions are omitted when they can be easily inferred. The operator $\otimes$ denotes the Kronecker product. Given matrices $A_1, A_2, \dots$ of sizes $r_1 \times r_1'$, $r_2 \times r_2', \dots$ the operator $\operatorname{blkdiag}(A_1, A_2, \dots)$ returns a $(r_1 + r_2 + \dots) \times (r_1' + r_2' + \dots)$ matrix B given by

$$B = \begin{bmatrix} A_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_2 & \mathbf{0} \\ & & \ddots \\ \mathbf{0} & & \end{bmatrix}.$$

We use indices i, j to refer to agents, and e to refer to edges. An edge e starting at agent i and ending at agent j is represented by $e = (i, j)$. Given a matrix A, the notation $[A]_{r,s}$ corresponds to the (r, s) entry of the matrix whereas $[A]_{(i,j)}$ or $[A]_e$ corresponds to the entry associated to the edge $e = (i, j)$. Given a matrix A, $\lambda(A)$ and $\rho(A)$ correspond respectively to its eigenvalues and to its spectral radius. We let agent 0 be the anchor node and use the superscript $+$ for data including information of the anchor.

II. PROBLEM DESCRIPTION

The localization problem consists of computing the planar poses $\{\mathbf{p}_0, \ldots, \mathbf{p}_n\}$ of $n+1 \in \mathbb{N}$ agents, where $\mathbf{p}_i = (x_i, y_i, \theta_i)$ for $i \in \{0, \ldots, n\}$, given $m \in \mathbb{N}$ noisy measurements of relative poses between agents. The measurements are arranged into a directed and connected graph $\mathcal{G} = (\mathcal{V}^+, \mathcal{E})$, where $\mathcal{V}^+ = \{0, \ldots, n\}$ and $|\mathcal{E}| = m$. There is an edge $e = (i, j) \in \mathcal{E}$ if node i has measured the pose $\mathbf{z}_{(i,j)}$ of node j in its own reference frame,

$$\mathbf{z}_{(i,j)} = \mathbf{p}_j \ominus \mathbf{p}_i + \mathbf{v}_{(i,j)}, \quad \mathbf{v}_{(i,j)} \sim N\left(\mathbf{0}, P_{(i,j)}\right), \tag{1}$$

where $\ominus$ is a standard pose composition operator,

$$\mathbf{p}_j \ominus \mathbf{p}_i \triangleq \left[\begin{array}{c} \left[\begin{array}{cc} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{array} \right]^T \left[\begin{array}{c} x_j - x_i \\ y_j - y_i \end{array} \right] \\ \theta_j - \theta_i \end{array} \right].$$

We let $\mathbf{z}^{xy}_{(i,j)}\mathbb{R}^2$ and $\mathbf{z}^{\theta}_{(i,j)} \in \mathbb{R}$ be the values in $\mathbf{z}_{(i,j)}$ associated to respectively the relative position and orientation, equivalently $P^{xy}_{(i,j)} \in \mathbb{R}^{2\times 2}$ and $P^{\theta}_{(i,j)} \in \mathbb{R}$ for the covariance matrix $P_{(i,j)}$. We assume that the measurements are independent since they were acquired individually by the agents. The $out-$neighbors of node i, $\mathcal{N}^+_{out}(i)$, are the nodes which i has observed, whereas its $in-$neighbors, $\mathcal{N}^+_{in}(i)$, are the ones that have measured i's pose,

$$\mathcal{N}^+_{in}(i) = \{j \in \mathcal{V}^+ | (j, i) \in \mathcal{E}\}, \quad \mathcal{N}^+_{out}(i) = \{j \in \mathcal{V}^+ | (i, j) \in \mathcal{E}\}.$$

The set of neighbors $\mathcal{N}^+(i)$ of i, is composed of both its $in-$ and $out-$neighbors, $\mathcal{N}^+(i) = \mathcal{N}^+_{in}(i) \cup \mathcal{N}^+_{out}(i)$, and we assume i can exchange data with all $j \in \mathcal{N}^+(i)$. We let $\mathcal{A}^+ \in \{0, 1, -1\}^{(n+1)\times m}$ be the incidence matrix of $\mathcal{G}$,

$$\mathcal{A}^+_{i,e} = \left\{ \begin{array}{ll} -1 & \text{if } e = (i,j) \\ 1 & \text{if } e = (j,i) \\ 0 & \text{otherwise} \end{array} \right., \text{ for } i \in \mathcal{V}^+, e \in \mathcal{E}, \tag{2}$$

and given any $r \in \mathbb{N}$ we let $\mathcal{A}^+_r$ be $\mathcal{A}^+ \otimes \mathbf{I}_r$.

The measurements in $\mathcal{G}$ must be properly combined to retrieve the positions $\{\mathbf{p}_0, \ldots, \mathbf{p}_n\}$ of the nodes in some reference frame. By convention, we let the first agent $i = 0$ be the origin of this reference frame, $\mathbf{p}_0 = (0, 0, 0)$, and refer to it by the anchor node. The goal is to compute the poses of the remaining nodes relative to the anchor. From now on, we let $\mathcal{V}$ be the set of non-anchor nodes,

$$\mathcal{V} = \{1, \ldots, n\} = \mathcal{V}^+ \setminus \{0\},$$

and $\mathcal{N}_{in}(i)$, $\mathcal{N}_{out}(i)$ and $\mathcal{N}(i)$ be defined as $\mathcal{N}^+_{in}(i)$, $\mathcal{N}^+_{out}(i)$ and $\mathcal{N}^+(i)$ but over $\mathcal{V}$ instead of $\mathcal{V}^+$. We let $\mathcal{A} \in \{0, 1, -1\}^{n\times m}$ be the result of deleting the row associated to the anchor node 0 from the incidence matrix $\mathcal{A}^+$, and equivalently $\mathcal{A}_r \in \{0, 1, -1\}^{rn\times rm}$ be $\mathcal{A} \otimes \mathbf{I}_r$.

We present a solution to the network localization problem that relies on a three-phases strategy:

Phase 1: Compute a suboptimal estimate of the node orientations $\hat{\theta} \in \mathbb{R}^n$ relative to node 0;

Phase 2: Express the position measurements of the nodes in terms of the previously computed orientations;

Phase 3: Compute the estimated positions and orientations of the nodes $\mathbf{p}^* = ((\mathbf{x}^*)^T, (\theta^*)^T)^T$.

III. MULTI-AGENT LOCALIZATION ALGORITHM

We begin by presenting a centralized algorithm that computes both the pose of the nodes and the associated covariance provided that all the measurements are available to a central node. We let the m measurements be arranged into the vector $\mathbf{z} \in \mathbb{R}^{3m}$ with associated covariance matrix $P_{\mathbf{z}} \in \mathbb{R}^{3m\times 3m}$, and we let $\mathbf{z}^{xy} \in \mathbb{R}^{2m}$, $P_{\mathbf{z}^{xy}} \in \mathbb{R}^{2m\times 2m}$ and $\mathbf{z}^{\theta} \in \mathbb{R}^m$, $P_{\mathbf{z}^{\theta}} \in \mathbb{R}^{m\times m}$ contain only the values associated to respectively positions and orientations.

1) Phase 1: Based exclusively on the orientation measurements $\mathbf{z}^{\theta} \in \mathbb{R}^m$ with covariance $P_{\mathbf{z}^{\theta}} \in \mathbb{R}^{m\times m}$, an initial estimate of the node orientations $\hat{\theta} \in \mathbb{R}^n$ relative to node 0 is obtained as follows. When the orientations measurements are considered alone and they belong to $\pm\frac{\pi}{2}$, the estimation problem becomes linear, $\mathbf{z}^{\theta} = \mathcal{A}^T\theta$. The Best Linear Unbiased Estimator for θ in the previous system is given by [16],

$$\hat{\theta} = P_{\hat{\theta}}\ \mathcal{A}\ P^{-1}_{\mathbf{z}^{\theta}}\ \mathbf{z}^{\theta}, \qquad P_{\hat{\theta}} = \left(\mathcal{A}\ P^{-1}_{\mathbf{z}^{\theta}}\ \mathcal{A}^T\right)^{-1}. \tag{3}$$

From now on, we let $\hat{\theta}^+ \in \mathbb{R}^{n+1}$ be an expansion of $\hat{\theta}$ containing the orientation of the anchor node, $\hat{\theta}^+ = (0, \hat{\theta})^T$.

2) Phase 2: Each relative position measurement $\mathbf{z}^{\theta}_{(i,j)}$ associated to the edge $e = (i, j)$, was originally expressed in the local coordinates of agent i. During the second phase, these measurements are transformed into a common orientation using the previously computed $\hat{\theta}^+$. We let $\hat{R}_i = R(\hat{\theta}_i) \in \mathbb{R}^{2\times 2}$ be the rotation matrix associated to the orientation $\hat{\theta}_i$ and $\hat{S}_i = S(\hat{\theta}_i) \in \mathbb{R}^{2\times 2}$ be its derivative,

$$\hat{R}_i = \left[\begin{array}{cc} \cos\hat{\theta}_i & -\sin\hat{\theta}_i \\ \sin\hat{\theta}_i & \cos\hat{\theta}_i \end{array} \right], \quad \hat{S}_i = \left[\begin{array}{cc} -\sin\hat{\theta}_i & \cos\hat{\theta}_i \\ -\cos\hat{\theta}_i & -\sin\hat{\theta}_i \end{array} \right],$$

for $i \in \mathcal{V}^+$; note that $\hat{\theta}_0 = 0$. Considering together all the edges $\mathcal{E} = \{(i_1, j_1), \ldots, (i_m, j_m)\}$, we build the following rotation matrix $\hat{R} = R(\hat{\theta}^+) \in \mathbb{R}^{2m\times 2m}$,

$$\hat{R} = \text{blkdiag}(\hat{R}_{i_1}, \ldots, \hat{R}_{i_m}), \tag{4}$$

and let $J \in \mathbb{R}^{2m\times n}$ be its jacobian with respect to $\hat{\theta}$,

$$[J]_{e,i} = \left\{ \begin{array}{ll} \hat{S}_i \mathbf{z}^{xy}_{(i,j)} & \text{if } e = (i,j) \\ 0 & \text{otherwise} \end{array} \right., \text{ for } e \in \mathcal{E}, i \in \mathcal{V}.$$

The updated measurements $\mathbf{w} \in \mathbb{R}^{2m+n}$ and a first-order propagation of the uncertainty $P_{\mathbf{w}}$ are

$$\begin{aligned} \mathbf{w} &= \left[\begin{array}{c} \tilde{\mathbf{z}}^{xy} \\ \hat{\theta} \end{array} \right] = \left[\begin{array}{cc} \hat{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{array} \right] \left[\begin{array}{c} \mathbf{z}^{xy} \\ \hat{\theta} \end{array} \right], \\ P_{\mathbf{w}} &= \left[\begin{array}{cc} \hat{R} & J \\ \mathbf{0} & \mathbf{I} \end{array} \right] \left[\begin{array}{cc} P_{\mathbf{z}^{xy}} & \mathbf{0} \\ \mathbf{0} & P_{\hat{\theta}} \end{array} \right] \left[\begin{array}{cc} \hat{R}^T & \mathbf{0} \\ J^T & \mathbf{I} \end{array} \right]. \end{aligned} \tag{5}$$

3) Phase 3: During the last phase, the positions of the nodes $\mathbf{x}^* \in \mathbb{R}^{2n}$ relative to node 0 are computed, and an improved version $\theta^* \in \mathbb{R}^n$ of the previous orientations $\hat{\theta}$ is obtained. Let $\mathbf{p}^* \in \mathbb{R}^{3n}$ contain both the positions and orientations of the agents,

$$\mathbf{p}^* = \left[\begin{array}{c} \mathbf{x}^* \\ \theta^* \end{array} \right] = P_{\mathbf{p}^*} B P^{-1}_{\mathbf{w}} \mathbf{w}, \quad P_{\mathbf{p}^*} = \left(B P^{-1}_{\mathbf{w}} B^T\right)^{-1}, \tag{6}$$

where $P_{\mathbf{w}}$, $\mathbf{w}$ are given by (5) and $B = \text{blkdiag}\left(\mathcal{A}_2, \mathbf{I}\right)$, being $\mathcal{A}_2$ equal to $\mathcal{A}_r$ with $r = 2$.

4) Centralized Algorithm: Considering the three phases together, the final agents' positions $\mathbf{x}^*$ and orientations θ^* are

$$\mathbf{x}^* = L^{-1}\mathcal{A}_2 Y_{\tilde{\mathbf{z}}^{xy}} \left(\mathbf{I} + JP_{\theta^*}J^T Y_{\tilde{\mathbf{z}}^{xy}} E\right) \hat{R}\ \mathbf{z}^{xy},$$
$$\theta^* = (\mathcal{A}P_{\mathbf{z}^\theta}^{-1}\mathcal{A}^T)^{-1}\mathcal{A}P_{\mathbf{z}^\theta}^{-1}\mathbf{z}^\theta + P_{\theta^*}J^T Y_{\tilde{\mathbf{z}}^{xy}} E\ \hat{R}\ \mathbf{z}^{xy}, \quad (7)$$

$$\text{where } Y_{\tilde{\mathbf{z}}^{xy}} = (\hat{R}P_{\mathbf{z}^{xy}}\hat{R}^T)^{-1}, \qquad E = \mathcal{A}_2^T L^{-1}\mathcal{A}_2 Y_{\tilde{\mathbf{z}}^{xy}} - \mathbf{I},$$
$$P_{\theta^*} = (P_{\hat{\theta}}^{-1} - J^T\ Y_{\tilde{\mathbf{z}}^{xy}}\ E\ J)^{-1}, \qquad L = \mathcal{A}_2 Y_{\tilde{\mathbf{z}}^{xy}}\mathcal{A}_2^T. \quad (8)$$

A full development of these expressions can be found in the Appendix. From (7), it can be seen that the computation of $\mathbf{x}^*$ and θ^* involves matrix inversions and other operations that require the knowledge of the whole system. Although a priori the proposed strategy would require a centralized implementation, in the next sections we show a proposal to carry out the computations in a distributed way.

IV. DISTRIBUTED COMPUTATION

In this section we present a distributed version of the previous centralized algorithm. Each node $i \in \mathcal{V}$ estimates its optimal pose $[\mathbf{x}^*]_i$, $[\theta^*]_i$ based on the measurements of its outgoing (i,j) and incoming (j,i) edges, and on local interactions with its neighbors $j \in \mathcal{N}(i)$. From now on, we let $Y^{xy}_{(i,j)}$, $Y^{\theta}_{(i,j)}$, and $\mathbf{i}^{xy}_{(i,j)}$, $\mathbf{i}^{\theta}_{(i,j)}$ be the information matrices and vectors of the relative measurements of each edge $(i,j) \in \mathcal{E}$,

$$Y^{xy}_{(i,j)} = (P^{xy}_{(i,j)})^{-1}, \qquad \mathbf{i}^{xy}_{(i,j)} = Y^{xy}_{(i,j)}\mathbf{z}^{xy}_{(i,j)},$$
$$Y^{\theta}_{(i,j)} = (P^{\theta}_{(i,j)})^{-1}, \qquad \mathbf{i}^{\theta}_{(i,j)} = Y^{\theta}_{(i,j)}\mathbf{z}^{\theta}_{(i,j)}. \quad (9)$$

1) Phase 1: The initial orientation $\hat{\theta}$ in the first phase of the algorithm can be computed in a distributed fashion using the Jacobi iterations [13]–[15] of eq. (3). Each agent $i \in \mathcal{V}$ maintains a variable $\hat{\theta}_i(t) \in \mathbb{R}$, initialized at $t = 0$ with $\hat{\theta}_i(0)$, and updated at each time step $t \in \mathbb{N}$ by

$$\hat{\theta}_i(t+1) = C_i^{-1}\left(\sum\nolimits_{j\in\mathcal{N}^+_{in}(i)}\mathbf{i}^{\theta}_{(j,i)} - \sum\nolimits_{j\in\mathcal{N}^+_{out}(i)}\mathbf{i}^{\theta}_{(i,j)}\right)$$
$$+ C_i^{-1}\left(\sum\nolimits_{j\in\mathcal{N}_{out}(i)}Y^{\theta}_{(i,j)}\hat{\theta}_j(t) + \sum\nolimits_{j\in\mathcal{N}_{in}(i)}Y^{\theta}_{(j,i)}\hat{\theta}_j(t)\right),$$
$$\text{where } C_i = \sum\nolimits_{j\in\mathcal{N}^+_{out}(i)}Y^{\theta}_{(i,j)} + \sum\nolimits_{j\in\mathcal{N}^+_{in}(i)}Y^{\theta}_{(j,i)}. \quad (10)$$

Let matrices $C, D \in \mathbb{R}^{m\times m}$ be $C = \text{blkdiag}(C_1, \ldots, C_n)$ and $D = C - P_{\hat{\theta}}^{-1}$. The previous Jacobi iterations converge if the spectral radius of $C^{-1}D$ is less than one,

$$\rho(C^{-1}D) < 1. \quad (11)$$

In addition, $\rho(C^{-1}D)$ gives the convergence speed of the system, converging faster for $\rho(C^{-1}D)$ closer to 0. Recalling that $P_{\mathbf{z}^\theta}$ is a diagonal matrix, then each variable $\hat{\theta}_i(t)$ asymptotically converges to the i-th entry $[\hat{\theta}]_i$ of the vector $\hat{\theta}$ of the centralized algorithm in (3), see [14].

Observe that the computations are fully distributed and they exclusively rely on local information. Each node $i \in \mathcal{V}$ exclusively uses the covariances $P^{\theta}_{(i,j)}$, $P^{\theta}_{(j,i)}$ and measurements $\mathbf{z}^{\theta}_{(i,j)}$, $\mathbf{z}^{\theta}_{(j,i)}$ of its incoming or outgoing edges $j \in \mathcal{N}^+(i)$ and the variables $\hat{\theta}_j(t)$ of its neighbors $j \in \mathcal{N}(i)$.

2) Phase 2: The agents execute $t_{\max}$ iterations of the previous algorithm. We let $\bar{\theta}_i$ be their orientation at iteration $t_{\max}$, $\bar{\theta}_i = \hat{\theta}_i(t_{\max})$ and $\bar{R} = R(\bar{\theta})$ be defined by using $\bar{\theta}_i$ instead of $\hat{\theta}_i$ in (4). The agents execute the second phase as in (5) to transform the position measurements $\mathbf{z}^{xy}$ into the common orientations $\bar{\theta}$; note that $\bar{\theta}$ does not change during this phase. Since matrix $\bar{R}$ is block diagonal, the transformation $\bar{R}\mathbf{z}^{xy}$ is local and it is completed in a single iteration. Each agent $i \in \mathcal{V}$ locally transforms its own local measurements,

$$\bar{\mathbf{z}}^{xy}_{(i,j)} = \bar{R}_i\ \mathbf{z}^{xy}_{(i,j)}, \quad \text{for } j \in \mathcal{N}^+_{out}(i). \quad (12)$$

Note that the agents use $\bar{\theta}$ instead of $\hat{\theta}$ and thus the updated measurements obtained are $\bar{\mathbf{z}}^{xy}$ instead of $\tilde{\mathbf{z}}^{xy}$.

3) Phase 3: In order to obtain the final poses, the third phase of the algorithm (6) apparently requires the knowledge of the covariance matrices $P_{\hat{\theta}}$, $P_{\mathbf{w}}$ of phases 1 and 2. However, a distributed computation of these matrices cannot be carried out in an efficient way. We show that due to the structure of the information matrices, the third phase of the algorithm can be expressed in terms of local data and can be executed in a distributed fashion. We let $\tilde{Y}^{xy}_{(i,j)}$ be the block within the matrix $Y_{\tilde{\mathbf{z}}^{xy}}$ in (8) associated to an edge $(i,j) \in \mathcal{E}$,

$$\tilde{Y}^{xy}_{(i,j)} = \hat{R}_i Y^{xy}_{(i,j)}\hat{R}_i^T,$$

and note that it can be locally computed by nodes i and j.

We let each agent $i \in \mathcal{V}$ maintain an estimate of its pose $\mathbf{p}_i(t) \in \mathbb{R}^3$ relative to the anchor, and we let $\mathbf{p}(t)$ be the result of putting together the $\mathbf{p}_i(t)$ variables for all $i \in \mathcal{V}$. Each node initializes its variable $\mathbf{p}_i(t)$ at $t = 0$ with $\mathbf{p}_i(0)$ and updates it at each time step $t \in \mathbb{N}$ by

$$\mathbf{p}_i(t+1) = M_i^{-1}\left(\mathbf{f}_i(\mathbf{p}(t)) + \mathbf{m}_i\right), \quad \text{where} \quad (13)$$

$$M_i = \begin{bmatrix} M_1 & M_2 \\ M_2^T & M_3 \end{bmatrix},\ \mathbf{f}_i(\mathbf{p}(t)) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix},\ \mathbf{m}_i = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}.$$

The elements within M_i are

$$M_1 = \sum\nolimits_{j\in\mathcal{N}^+_{out}(i)}\tilde{Y}^{xy}_{(i,j)} + \sum\nolimits_{j\in\mathcal{N}^+_{in}(i)}\tilde{Y}^{xy}_{(j,i)},$$
$$M_2 = \sum\nolimits_{j\in\mathcal{N}^+_{out}(i)}\tilde{Y}^{xy}_{(i,j)}\hat{S}_i\mathbf{z}^{xy}_{(i,j)},$$
$$M_3 = \sum\nolimits_{j\in\mathcal{N}^+_{out}(i)}\left((\mathbf{z}^{xy}_{(i,j)})^T\hat{S}_i^T\tilde{Y}^{xy}_{(i,j)}\hat{S}_i\mathbf{z}^{xy}_{(i,j)} + Y^{xy}_{(i,j)}\right)$$
$$+ \sum\nolimits_{j\in\mathcal{N}^+_{in}(i)}Y^{xy}_{(j,i)}. \quad (14)$$

The elements within $\mathbf{f}_i(\mathbf{p}(t))$, which is the term depending on the previous estimates $\mathbf{p}(t)$, are

$$f_1 = \sum_{j\in\mathcal{N}_{out}(i)}\tilde{Y}^{xy}_{(i,j)}\mathbf{x}_j(t) + \sum_{j\in\mathcal{N}_{in}(i)}\tilde{Y}^{xy}_{(j,i)}\left(\mathbf{x}_j(t) + \hat{S}_j\mathbf{z}^{xy}_{(j,i)}\theta_j(t)\right),$$
$$f_2 = \sum\nolimits_{j\in\mathcal{N}_{out}(i)}(\mathbf{z}^{xy}_{(i,j)})^T\hat{S}_i^T\tilde{Y}^{xy}_{(i,j)}\mathbf{x}_j(t)$$
$$- \sum\nolimits_{j\in\mathcal{N}_{out}(i)}Y^{xy}_{(i,j)}\theta_j(t) - \sum\nolimits_{j\in\mathcal{N}_{in}(i)}Y^{xy}_{(j,i)}\theta_j(t). \quad (15)$$

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

$$
\begin{aligned}
m_1 &= \sum\nolimits_{j\in\mathcal{N}^+_{in}(i)} \tilde{Y}^{xy}_{(j,i)} \left(\tilde{\mathbf{z}}^{xy}_{(j,i)} - \hat{S}_j \mathbf{z}^{xy}_{(j,i)} \hat{\theta}_j\right) \\
&- \sum\nolimits_{j\in\mathcal{N}^+_{out}(i)} \tilde{Y}^{xy}_{(i,j)} \left(\tilde{\mathbf{z}}^{xy}_{(i,j)} - \hat{S}_i \mathbf{z}^{xy}_{(i,j)} \hat{\theta}_i\right) \\
m_2 &= \sum_{j\in\mathcal{N}^+_{in}(i)} Y^{\theta}_{(j,i)} \hat{\theta}_i - \sum_{j\in\mathcal{N}_{out}(i)} Y^{\theta}_{(i,j)} \hat{\theta}_j - \sum_{j\in\mathcal{N}_{in}(i)} Y^{\theta}_{(j,i)} \hat{\theta}_j \\
&+ \sum_{j\in\mathcal{N}^+_{out}(i)} \left((\mathbf{z}^{xy}_{(i,j)})^T \hat{S}_i^T \tilde{Y}^{xy}_{(i,j)} \left(\hat{S}_i \mathbf{z}^{xy}_{(i,j)} \hat{\theta}_i - \tilde{\mathbf{z}}^{xy}_{(i,j)}\right) + Y^{\theta}_{(i,j)} \hat{\theta}_i \right).
\end{aligned}
\quad (16)
$$

***Theorem** 4.1:* The estimates $\mathbf{p}_i(t)$ computed by each node $i \in \mathcal{V}$ by the distributed algorithm (13) converge to $\mathbf{p}_i^* = \left((\mathbf{x}_i^*)^T, \theta_i^*\right)^T$ for connected measurement graphs $\mathcal{G}$ with ring or string structure.

Proof: Eqs. (13)-(16) are the Jacobi iterations at agent $i \in \mathcal{V}$ of the third phase of the localization algorithm,

$$Y_{\mathbf{p}^*} \ \mathbf{p}^* = B \ P_{\mathbf{w}}^{-1} \ \mathbf{w}, \quad (17)$$

being $Y_{\mathbf{p}^*}$, $B \ P_{\mathbf{w}}^{-1}$, and $\mathbf{w}$ given in the appendix. Let $M = \mathrm{blkdiag}(M_1, \ldots, M_n)$ and $\mathbf{q}^*$ be a permutation of $\mathbf{p}^*$ so that the estimates of each node appear together, $\mathbf{q}^* = \left((\mathbf{x}_1^*)^T, \theta_1^*, \ldots, (\mathbf{x}_n^*)^T, \theta_n^*\right)^T$. Equivalently, the permuted version of $Y_{\mathbf{p}^*}$ is $Y_{\mathbf{q}^*}$. The estimates $\mathbf{p}_i(t)$ computed by each node $i \in \mathcal{V}$ with the distributed algorithm (13) converge to $\mathbf{p}_i^* = \left((\mathbf{x}_i^*)^T, \theta_i^*\right)^T$ if

$$\rho(\mathbf{I} - M^{-1} Y_{\mathbf{q}^*}) < 1. \quad (18)$$

Since $\lambda(\mathbf{I} - M^{-1} Y_{\mathbf{q}^*}) = 1 - \lambda(M^{-1} Y_{\mathbf{q}^*})$, then (18) is equivalent to $0 < \lambda(M^{-1} Y_{\mathbf{q}^*}) < 2$.

(i) $0 < \lambda(M^{-1} Y_{\mathbf{q}^*})$ can be easily checked taking into account that both M^{-1} and $Y_{\mathbf{q}^*}$ are nonsingular, symmetric, positive definite, and that $\lambda(M^{-1} Y_{\mathbf{q}^*}) \geq \frac{\lambda_{\min}(M^{-1})}{\lambda_{\max}(Y_{\mathbf{q}^*})}$ [17, Lemma 1]. Since $0 < \frac{\lambda_{\min}(M^{-1})}{\lambda_{\max}(Y_{\mathbf{q}^*})}$, then $0 < \lambda(M^{-1} Y_{\mathbf{q}^*})$.

(ii) In order to prove $\lambda(M^{-1} Y_{\mathbf{q}^*}) < 2$, we first focus on the structure of the information matrix $Y_{\mathbf{q}^*}$. This matrix has zeros for the elements associated to non neighboring nodes, 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. For ring or string graphs, the adjacency matrix $\mathrm{adj}(\mathcal{G})$ can be reordered grouping the elements around the main diagonal resulting in a matrix that has semi bandwidth $s = 1$, i.e.,

$$[\mathrm{adj}(\mathcal{G})]_{i,j} = 0 \text{ for } |i - j| > s. \quad (19)$$

As a consequence, the information matrix $Y_{\mathbf{q}^*}$ has block semi bandwidth $s' = 1$, and as stated by [17, Theorem 1],

$$\lambda_{\max}(M^{-1} Y_{\mathbf{q}^*}) < 2^{s'} = 2. \quad (20)$$

■

Due to the structure of the information matrices, the third phase of the algorithm can be expressed in terms of local information (13)-(16) and interactions with neighbors, and thus it can be implemented in a distributed fashion. It is observed that the agents actually use $\bar{\theta}$ instead of $\hat{\theta}$ 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.

V. SIMULATIONS

A set of simulations have been carried out to show the performance of the algorithm and to compare the results of the distributed (Section IV) and the centralized (Section III) approaches.

In the first set of experiments, a team of 20 agents are placed along a ring of radius 4 m with their orientations randomly generated within $\pm\frac{pi}{2}$ (Fig. 1). Each agent measures the relative pose of the next agent and these measurements are corrupted with noises in the x- and y- coordinates of 6 cm standard deviation, and of 1 degree for the orientation. The agents execute the proposed method to compute their pose with respect to the anchor node $R1$. The experiment

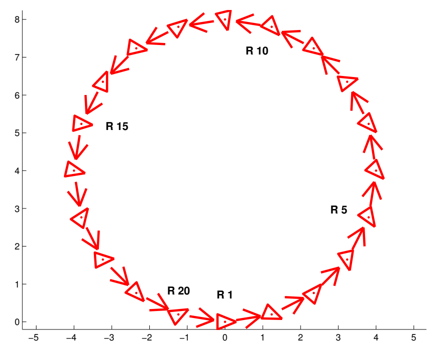


Fig. 1. A team of 20 agents are placed along a circle of $4m$ radius. Each agent (red triangles) measures the relative pose of its next agent (outgoing arrow), and exchanges data with both its previous and next agents.

is repeated 100 times and the average results can be seen in Table I. The centralized solution (Section III) presents a high accuracy. The greatest difference between the ground truth and the obtained positions are of around 25 cm for the $x-$ and $y-$ coordinates, with around 13 cm standard deviation, and of around 4 degrees for the orientation, with approximately 1.7 degrees of standard deviation. The

TABLE I

RESULTS FOR THE SCENARIO IN FIG. 1

CENTRALIZED ALGORITHM VS. GROUND TRUTH			
	Max error	Avg 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)			
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°

distributed algorithm (Section IV) has also been tested for the scenario in Fig.1 and its results have been compared to the ones obtained by the centralized algorithm (Section III). We use the flagged initialization [14] 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 (11) and $\rho(\mathbf{I} - M^{-1} Y_{\mathbf{q}^*})$ in (18). The closer to one the values are, the slower the system converges. Observe that the second phase is always executed in a single iteration and for this reason it does not have any convergence speed associated. In this experiment, both $\rho(C^{-1}D)$ and $\rho(\mathbf{I} - M^{-1}Y_{\mathbf{q}^*})$ are close to one (=0.99), and thus we expect the algorithm to converge slowly. After executing the first phase during $t = 50$ iterations, the obtained $\bar{\theta}$ still differs from the centralized solution $\hat{\theta}$ 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 $\bar{\theta}$ instead of $\hat{\theta}$, the final results also differ. For the case $t = 200$ (third column), the difference between the pose estimated by the distributed and centralized approaches is small (1.64 cm and 0.48 cm for the $x-$ and $y-$ coordinates, and 0.11 degrees for the orientation), and similar results are obtained for $t = 100$. However, for $t = 50$ the final errors are larger (1.74 cm and 0.84 cm for the $x-$ and $y-$ coordinates, and 0.29 degrees for the orientation), which is due to the use of a less accurate approximation of $\hat{\theta}$.

A simulation with 10 agents placed as in Fig. 2 has been carried out. An agent i measures the relative position and orientation of a second agent j if there is an arrow from i into j. The measured positions are corrupted with additive noises of standard deviation proportional to the distance d between the nodes, 5% d for the x-coordinate, and 0.7% d for the y-coordinate. And the measured orientations are corrupted with additive noises of 2.5 degrees of standard deviation.

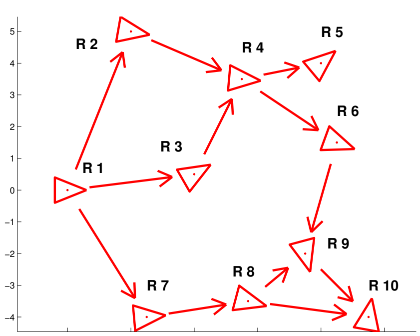


Fig. 2. A team of 10 agents are placed at unknown poses in a planar environment. Each agent can measure the relative pose (position and orientation) of a subset of the other agents (outgoing arrows).

The agents execute the distributed algorithm (Fig. 4) during the phase 1 to compute their orientations with respect to the anchor node R1. At $t = 0$ they initialize their orientations to 0 (black). During the iterations the agents update their estimates (gray), which successively approach the orientations that would be obtained by a centralized system (blue). After the first phase (Fig. 3(a)), the orientations computed by the agents (blue) are very close to the ground truth data (red). After that, they execute the second phase to transform their local measurements. The relative positions measured by the

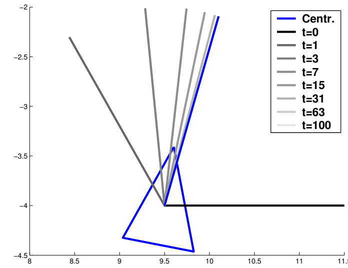


Fig. 4. Phase 1 of the proposed strategy. The orientations estimated by the agent R10 along different iterations of the distributed algorithm (gray) successively approach the centralized solution (blue).

agents were initially expressed into the reference frame of the observer. After the phase 2, they are expressed in the reference frame of the anchor node (Fig. 3(b)). And finally, they execute the last phase to obtain both their positions and orientations relative to the anchor node (Fig. 3(c)). As an example, a detail of the evolution of the estimates at R10 can be observed in Fig. 5. Although the convergence

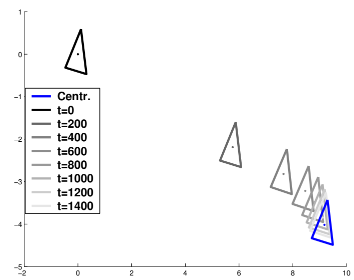
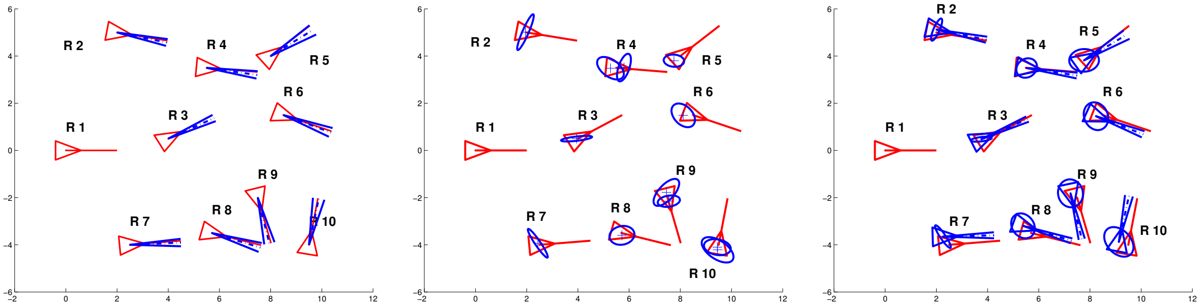


Fig. 5. At $t = 0$ the estimated pose of R10 in Fig. 2 is initialized in the origin (black). Along the iterations, its estimated orientation almost does not vary, while its pose (gray) successively approaches the the centralized solution (blue).

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.

VI. CONCLUSIONS

In this paper, a strategy for computing the poses of an agent network based on relative noisy measurements has been presented. It relies on the decomposition of the original problem into different phases that can be solved by linear estimation methods. A solution that computes both the poses and the covariances but that requires a centralized computation has been presented. In addition, a distributed solution that computes the poses has also been presented and its convergence has been proved for certain classes of communication graphs. The proposed algorithms have been experimentally evaluated and their performance for solving the localization problem has been demonstrated. An interesting result of the simulations is that the distributed approach converged under a wider amount of classes of communication graphs than the ones considered in the theoretical analysis. As future work we plan to carry out a deeper study of its convergence properties.



(a) Phase 1 (b) Phase 2 (c) Phase 3

Fig. 3. The agents execute the proposed strategy to obtain their poses relative to the anchor R1 for the experiment in Fig. 2. The estimates (blue dashed) are compared to the ground truth data (red solid). The covariances computed by the centralized approach are also displayed (blue solid). (a) After the first phase, the agents obtain their orientations relative to the anchor R1 (blue dashed). These orientation are very similar to the ground data (red solid). (b) The agents transform their local position measurements according to the previously obtained global orientation. (c) Finally, the agents compute their position relative to the anchor (blue triangles). We are also displaying the covariances obtained by the centralized algorithm (blue ellipses).

ACKNOWLEDGMENT

This work was supported by projects Ministerio de Ciencia e Innovacion DPI2009-08126 and Italian Ministry of University and Research (MIUR) MEMONET National Research Project, and grants MEC BES-2007-14772 and PRIN 20087W5P2K.

REFERENCES

[1] G. Calafiore, L. Carlone, and M. Wei, "A distributed gradient method for localization of formations using relative range measurements," in *IEEE Multi-Conf. on Systems and Control*, Yokohama, Japan, Sep. 2010, accepted.

[2] ——, "A distributed gauss-newton approach for range-based localization of multi agent formations," in *IEEE Multi-Conf. on Systems and Control*, Yokohama, Japan, Sep. 2010, accepted.

[3] A. Sarlette, R. Sepulchre, and N. E. Leonard, "Autonomous rigid body attitude synchronization," *Automatica*, vol. 45, no. 2, pp. 572–577, 2008.

[4] M. Franceschelli and A. Gasparri, "On agreement problems with gossip algorithms in absence of common reference frames," in *IEEE Int. Conf. on Robotics and Automation*, Anchorage, USA, May 2010, pp. 4481–4486.

[5] J. Cortes, "Global and robust formation-shape stabilization of relative sensing networks," *Automatica*, vol. 45, no. 12, pp. 2754 – 2762, 2009.

[6] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693 –703, 2007.

[7] G. Lafferriere, A. Williams, J. Caughman, and J. J. P. Veerman, "Decentralized control of vehicle formations," *Systems & Control Letters*, vol. 54, no. 9, pp. 899 – 910, 2005.

[8] S. Roumeliotis and G. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[9] N. Trawny, S. I. Roumeliotis, and G. B. Giannakis, "Cooperative multi-robot localization under communication constraints," in *IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, May 2009, pp. 4394–4400.

[10] X. Zhou and S. Roumeliotis, "Robot-to-robot relative pose estimation from range measurements," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1379–1393, 2008.

[11] C. Sagues, A. C. Murillo, J. J. Guerrero, T. Goedemé, T. Tuytelaars, and L. V. Gool, "Localization with omnidirectional images using the 1d radial trifocal tensor," in *Proc of the IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 551–556.

[12] N. Trawny, X. S. Zhou, K. X. Zhou, and S. I. Roumeliotis, "Inter-robot transformations in 3-d," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 226–243, 2010.

[13] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*.

[14] P. Barooah and J. Hespanha, "Distributed estimation from relative measurements in sensor networks," in *Int. Conf. on Intelligent Sensing and Information Processing*, Chennai, India, Jan. 2005, pp. 88–93.

[15] ——, "Estimation on graphs from relative measurements," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 57–74, 2007.

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

[17] L. Kolotilina, "Bounds for eigenvalues of symmetric block jacobi scaled matrices," *Journal of Mathematical Sciences*, vol. 79, no. 3, pp. 1043–1047, 1996.

APPENDIX

Consider that $\hat{\theta}$ and its covariance $P_{\hat{\theta}}$ are as in eq. (3). The updated measurements $\mathbf{w}$ and covariance $P_{\mathbf{w}}$ in eq. (5) can be expressed as

$$\mathbf{w} = \begin{bmatrix} \hat{R}\mathbf{z}^{xy} \\ \hat{\theta} \end{bmatrix}, \; P_{\mathbf{w}} = \begin{bmatrix} \hat{R}P_{\mathbf{z}^{xy}}\hat{R}^T + JP_{\hat{\theta}}J^T & P_{\hat{\theta}}J^T \\ JP_{\hat{\theta}} & P_{\hat{\theta}} \end{bmatrix}. \quad (21)$$

In order to develop an expression for x^* and θ^* in eq. (6), it is necessary to invert matrix $P_{\mathbf{w}}$, which using classical blockwise inversion relations gives

$$(P_{\mathbf{w}})^{-1} = \begin{bmatrix} Y_{\tilde{\mathbf{z}}^{xy}} & -Y_{\tilde{\mathbf{z}}^{xy}}J \\ -J^T Y_{\tilde{\mathbf{z}}^{xy}} & P_{\hat{\theta}}^{-1} + J^T Y_{\tilde{\mathbf{z}}^{xy}} J \end{bmatrix}, \quad (22)$$

with $Y_{\tilde{\mathbf{z}}^{xy}}$ as in eq. (8). Then, $(P_{\mathbf{w}})^{-1}$ can be used to obtain $P_{\mathbf{p}^*}$ in eq. (6). For simplicity, we first compute its information matrix $Y_{\mathbf{p}^*} = (P_{\mathbf{p}^*})^{-1}$,

$$Y_{\mathbf{p}^*} = \begin{bmatrix} \mathcal{A}_2 Y_{\tilde{\mathbf{z}}^{xy}} \mathcal{A}_2^T & -\mathcal{A}_2 Y_{\tilde{\mathbf{z}}^{xy}} J \\ -J^T Y_{\tilde{\mathbf{z}}^{xy}} \mathcal{A}_2^T & P_{\hat{\theta}}^{-1} + J^T Y_{\tilde{\mathbf{z}}^{xy}} J \end{bmatrix}. \quad (23)$$

We invert $Y_{\mathbf{p}^*}$ and get $P_{\mathbf{p}^*} = \begin{bmatrix} P_{\mathbf{x}^*} & P_{\mathbf{x}^*,\theta^*} \\ P_{\mathbf{x}^*,\theta^*}^T & P_{\theta^*} \end{bmatrix}$, with

$$\begin{aligned} P_{\theta^*} &= \left(P_{\hat{\theta}}^{-1} - J^T Y_{\tilde{\mathbf{z}}^{xy}} E J\right)^{-1}, \\ P_{\mathbf{x}^*} &= L^{-1} + L^{-1}\mathcal{A}_2 Y_{\tilde{\mathbf{z}}^{xy}} J P_{\theta^*} J^T Y_{\tilde{\mathbf{z}}^{xy}} \mathcal{A}_2^T L^{-1}, \\ P_{\mathbf{x}^*,\theta^*} &= L^{-1}\mathcal{A}_2 Y_{\tilde{\mathbf{z}}^{xy}} J P_{\theta^*}, \end{aligned} \quad (24)$$

being $Y_{\tilde{\mathbf{z}}^{xy}}$, E and L as in (8). Combining the expressions for $\mathbf{w}$ and $P_{\mathbf{p}^*}$ in eqs. (21) and (24) with eq. (6) and taking into account that

$$BP_{\mathbf{w}}^{-1} = \begin{bmatrix} \mathcal{A}_2 Y_{\tilde{\mathbf{z}}^{xy}} & -\mathcal{A}_2 Y_{\tilde{\mathbf{z}}^{xy}} J \\ -J^T Y_{\tilde{\mathbf{z}}^{xy}} & AP_{\mathbf{z}^{\theta}}^{-1}\mathcal{A}^T + J^T Y_{\tilde{\mathbf{z}}^{xy}} J \end{bmatrix}, \quad (25)$$

we obtain $\mathbf{x}^*$ and θ^* as in eq. (7).