A Centralized-equivalent Decentralized Implementation of Extended Kalman Filters for Cooperative Localization

Solmaz S. Kia Stephen F. Rounds Sonia Martínez
appeared in Proceeding of IROS 2014

Abstract— **We present a novel decentralized cooperative localization algorithm for mobile robots. The proposed algorithm is a decentralized implementation of a centralized Extended Kalman Filter for cooperative localization. In this algorithm, instead of propagating cross-covariance terms, each robot propagates new intermediate local variables that can be used in an update stage to create the required propagated cross-covariance terms. Whenever there is a relative measurement in the network, the algorithm declares the robot making this measurement as the *interim master*. By acquiring information from the *interim landmark*, the robot the relative measurement is taken from, the interim master can calculate and broadcast a set of intermediate variables which each robot can then use to update its estimates to match that of a centralized Extended Kalman Filter for cooperative localization. Once an update is done, no further communication is needed until the next relative measurement. The communication graph can be a time-varying directed graph with the only requirement that it should have a spanning tree rooted at the interim master.**

I. INTRODUCTION

The successful deployment of multi-robot systems in tasks such as search and rescue, environmental monitoring, and oceanic exploration depends on the accurate localization of these robots. In these applications, the environment is often uncharted, dynamic, and may not be accessible a priori. Thus, classical beacon-based localization algorithms [1] or fixed feature-based Simultaneous Localization and Mapping algorithms [2] may not be applicable. Fully or intermittently GPS-denied environments also deprive these applications from exploiting GPS navigation. A technique that can work best for such multi-robot systems is a *Cooperative Localization* (CL) strategy. This technique uses relative measurements among the robots as a feedback signal to jointly estimate the location of team members, resulting in increased position accuracy for the entire team. However, the real benefit of CL is when occasional access to accurate absolute localization information is available to some members, which then is spread to other team members by means of CL. In this paper, we present a novel decentralized CL algorithm.

Available CL algorithms are either centralized or decentralized. Although centralized schemes (see e.g., [3], [4]) result in less conservative estimations, their lack of robustness and energy inefficiency make them less preferable. A major challenge in developing a decentralized CL (D-CL) algorithm is how to keep an accurate account of all cross-correlations among robots without all-to-all communication at each time-step. Ignoring these cross-correlations in future updates results in overconfidence in pose estimates that can lead to divergence of the estimates. Also, keeping track of cross-correlations benefits further the entire team from updating relative measurements between any two members. In [5], a suboptimal algorithm–where only the robot obtaining the relative measurement updates its states–is proposed where, in order to produce consistent estimates, a bank of Extended Kalman Filters (EKFs) is maintained at each robot. Using an accurate book-keeping of the identity of the robots involved in previous updates and the age of such information, each of these filters is only updated when its propagated state is not correlated to the state involved in the current update equation. The computational complexity, the large memory demand, and the growing size of information needed at each update time are the main drawbacks of this algorithm. An alternative approach to develop D-CL algorithms is to distribute the computation of components of a centralized CL among team members. In a straightforward fashion, this decentralization can be conducted as a multi-centralized CL, wherein each robot broadcasts its own information to the entire team. Then, every robot can calculate and reproduce the centralized pose estimates, i.e., each robot acts as a Fusion Center (FC) [6]. Besides a high-processing cost for each robot, this scheme requires all-to-all robot communication at the time of each information exchange. A D-CL algorithm distributing computations of an EKF centralized CL algorithm is proposed in [7]. To decentralize the cross-covariance propagation, [7] uses a singular-value decomposition to split each cross-covariance term between the corresponding two robots. Then, each robot propagates its portion. However, at update times, the separated parts should be combined, requiring an all-to-all robot communication in the correction step. Subsequently, [8] presents a maximum-a-posteriori (MAP) D-CL algorithm in which all the robots in the team calculate parts of the centralized CL. A D-CL approach equivalent to a centralized CL is proposed

This work was supported by L3 Communications through the UCSD Cymer Center for Control Systems and Dynamics. This presentation consists of general capabilities information that does not contain controlled technical data as defined within the International Traffic in Arms (ITAR) Part 120.10 or Export Administration Regulations (EAR) Part 734.7-11. The first author is with the Department of Mechanical and Aerospace Engineering, University of California Irvine, Irvine, CA 92697, USA, solmaz@uci.edu, the second author is with John Deere ISG – Torrance, 20780 Madrona Avenue, Torrance, CA 90503, USA, RoundsStephenF@JohnDeere.com, and the third author is with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA 92093, USA, soniamd@ucsd.edu

in [9]. This scheme handles both limited communication ranges and time-varying communication graphs by using an information transfer scheme wherein each robot broadcasts all its locally available information to every robot within its communication radius at each time-step. The broadcasted information of each robot includes the past and present measurements, as well as past measurements previously received from other robots. The main drawback of this method is its high communication cost, which may not be affordable in applications with limited communication bandwidth. Finally, CL techniques to handle system and measurement models with non-Gaussian noises are discussed in [10], [11].

In this note, we propose a novel recursive D-CL algorithm called *Interim Master D-CL* which is exactly equivalent to the centralized EKF for CL of [7]. Our algorithm is developed by using new intermediate variables that eliminate the explicit calculation of the cross-covariance terms, resulting in decoupled propagation equations. The update stage is performed by designating the robot making the relative measurement as the *interim master*, which provides the rest of the robots with the information they need to update their pose and covariance in a manner that exactly matches those of a centralized EKF for CL. In particular, the size of the associated messages is independent of the size of the team. To calculate the update equations, the interim master only requires information from the *interim landmark*, the robot that the relative measurement is taken from. Because the propagation stage is fully decoupled, if there is no relative measurement in the network, no intra-network communication is needed. The communication graph can be a time-varying directed graph with the only requirement that it should have a spanning tree rooted at the interim master, see Fig. 1. Our algorithm can easily incorporate absolute measurements, and is robust to permanent robot drop-outs.

II. Preliminaries

In this section, we introduce our notation, terminology, and the description of the mobile robotic group we study.

A. Notation and communication graph terminology

Let $\mathbb{R}$ denote the set of real numbers and $\mathbb{M}_n$ represent the set of real positive definite matrices of dimension $n\times n$. We denote by $\mathbf{0}_{n\times m}$ the zero matrix of dimension $n\times m$, and by $\mathbf{I}_n$ the identity matrix of dimension $n\times n$. The transpose of matrix $\mathbf{A}\in\mathbb{R}^{n\times m}$ is $\mathbf{A}^\top$. We use $\mathrm{Diag}(\mathbf{A}_1,\cdots,\mathbf{A}_N)$ to denote the block-diagonal matrix constructed from matrices $\mathbf{A}_1,\dots,\mathbf{A}_N$. For finite sets V_1 and V_2, we denote by $V_1\backslash V_2$ the set whose elements consist of all the elements of V_1 that are not in V_2. We distinguish the variables associated to robot i by the superscript i, e.g., $\mathbf{x}^i$ is the pose (i.e., position and orientation) of robot i, $\hat{\mathbf{x}}^i$ is its pose estimate, and $\mathbf{P}^i$ is the covariance matrix of its pose estimate. The cross-covariance[1] of the pose vectors of robots i and j is

[1] In this note, we use the term *cross-covariance* to refer to the correlation terms between two robots in the covariance matrix of the entire network.

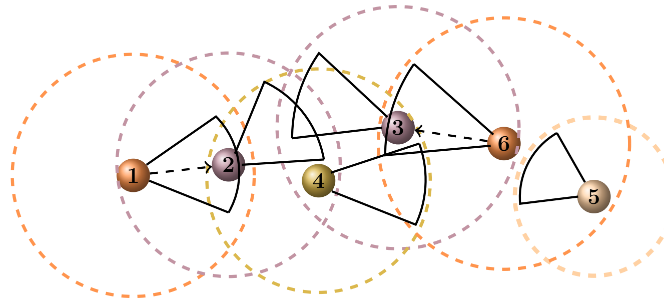

Fig. 1: The spheres represent the robots and the dashed circles around them represents the communication range of the robots. The circular sectors depict the exteroceptive sensing zone of the corresponding robot. Here, robots 1 and 6 make relative measurements, respectively, of robots 2 and 3. For each of robots 1 and 6, there is a spanning tree in the communication graph of this team that is rooted at these robots.

$\mathbf{P}_{ij}$. We denote the propagated and updated variables, say $\hat{\mathbf{x}}^i$, at time-step k by $\hat{\mathbf{x}}^{i-}(k)$ and $\hat{\mathbf{x}}^{i+}(k)$, respectively. We drop the time-step argument of the variables whenever it is clear from the context. If $\mathbf{q}^i\in\mathbb{R}^{n^i}$ is a local variable at robot i in a network of N robots, the aggregated $\mathbf{q}^i$'s is represented by $\mathbf{q}=(\mathbf{q}^1,\dots,\mathbf{q}^N)\in\mathbb{R}^d$, $d=\sum_{i=1}^N n^i$. Finally, we define our communication graph terminology, see e.g. [12]. A *directed graph* is a pair $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{1,\dots,N\}$ is the *node set* and $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$ is the *edge set*. An edge from i to j, depicted by an arrow from i to j, means that agent i can send information to agent j. A *directed path* is a sequence of consecutive nodes connected by edges. A *spanning tree* of a directed graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ is a subgraph $\mathcal{G}'=(\mathcal{V},\mathcal{E}')$ such that $\mathcal{E}'\subseteq\mathcal{E}$ and there is a *root* node in $\mathcal{V}$ connected to every other node in $\mathcal{G}'$ through unique directed paths.

B. Description of the mobile robot group

We consider a team of N mobile robots with processing and communication capabilities. Every robot has a distinct detectable identity. Every robot carries a proprioceptive sensor to measure its self motion and exteroceptive sensing devices to monitor the environment for localization features in its measurement range, which here are other robots in the team. Exteroceptive sensors can uniquely identify other robots in the team and measure relative pose, range, bearing or a combination of them. Every robot has a bounded communication range, see Fig. 1, and the communications happen in multi-hop fashion, i.e., every robot re-broadcasts every received message intended to reach the entire team. The motion of each robot is described by its own linear or nonlinear equations of motion. The collective motion equation of the team is given by:

$$\mathbf{x}(k+1)=\mathbf{f}(\mathbf{x}(k),\mathbf{u}(k))+\mathbf{g}(\mathbf{x}(k))\mathbf{n}(k), \tag{1}$$

where $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{n}$ are, respectively, the aggregated vectors of the pose $\mathbf{x}^i\in\mathbb{R}^{n^i}$, the input $\mathbf{u}^i\in\mathbb{R}^{m^i}$ and the process noise $\mathbf{n}^i\in\mathbb{R}^{p^i}$, $i\in\mathcal{V}$. Here, $\mathbf{f}(\mathbf{x},\mathbf{u})=(\mathbf{f}^1(\mathbf{x}^1,\mathbf{u}^1),\cdots,\mathbf{f}^N(\mathbf{x}^N,\mathbf{u}^N))$ and $\mathbf{g}(\mathbf{x})=\mathrm{Diag}(\mathbf{g}^1(\mathbf{x}^1),\cdots,\mathbf{g}^N(\mathbf{x}^N))$, where, $\mathbf{f}^i(\mathbf{x}^i,\mathbf{u}^i)$ and $\mathbf{g}^i(\mathbf{x}^i)$, are, respectively, the system function and process noise coefficient function of the robot $i \in \mathcal{V}$. We assume that the process noises $\mathbf{n}^i$, $i \in \mathcal{V}$, are independent zero-mean white Gaussian processes with a known variance $\mathbf{Q}^i = E[\mathbf{n}^{i\top}\mathbf{n}^i]$. We model the relative measurement collected by robot i from robot j as:

$$\mathbf{z}_{ij}(k+1) = \mathbf{h}_{ij}(\mathbf{x}^i(k), \mathbf{x}^j(k)) + \boldsymbol{\nu}^i(k), \quad \mathbf{z}_{ij} \in \mathbb{R}^{n_z^i}, \quad (2)$$

where $\mathbf{h}_{ij}(\mathbf{x}^i, \mathbf{x}^j)$ is the measurement model and $\boldsymbol{\nu}^i$ is the measurement noise of robot $i \in \mathcal{V}$, assumed to be independent zero-mean white Gaussian processes with known covariance $\mathbf{R}^i = E[\boldsymbol{\nu}^{i\top}\boldsymbol{\nu}^i]$. All sensor noises are assumed to be white and mutually uncorrelated. We show below how using an EKF, relative measurements between robots are used to improve the propagated states of the system. Here, we assume that all the sensor measurements are synchronized.

III. Benchmark Centralized Cooperative Localization Algorithm

In this section, we revisit the centralized EKF CL algorithm of [7] as our benchmark solution. Our main contribution, presented in the next section, is to offer a novel decomposition of the computations of this algorithm which results in a decentralized implementation without the need to an all-to-all communication.

Centralized EKF CL Initialization

For $i \in \mathcal{V}$, we initialize the EKF algorithm at:

$$\hat{\mathbf{x}}^{i+}(0) \in \mathbb{R}^{n^i}, \ \mathbf{P}^{i+}(0) \in \mathbb{M}_{n^i}, \mathbf{P}^{+}_{ij}(0) = \mathbf{0}_{n^i \times n^j}, \ j \in \mathcal{V}\backslash\{i\}.$$

Centralized EKF CL Propagation

Using the collective motion model (1), the collective EKF state and covariance propagation equations are:

$$\hat{\mathbf{x}}^{-}(k+1) = \mathbf{f}(\hat{\mathbf{x}}^{+}(k), \mathbf{u}(k)), \quad (3a)$$
$$\mathbf{P}^{-}(k+1) = \mathbf{F}(k)\mathbf{P}^{+}(k)\mathbf{F}(k)^\top + \mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}(k)^\top, \quad (3b)$$

where $\mathbf{F} = \text{Diag}(\mathbf{F}^1, \cdots, \mathbf{F}^N)$, $\mathbf{G} = \text{Diag}(\mathbf{G}^1, \cdots, \mathbf{G}^N)$ and $\mathbf{Q} = \text{Diag}(\mathbf{Q}^1, \cdots, \mathbf{Q}^N)$, with, for all $i \in \mathcal{V}$, $\mathbf{F}^i = \frac{\partial}{\partial \mathbf{x}^i}\mathbf{f}(\hat{\mathbf{x}}^{i+}(k), \mathbf{u}^i(k))$ and $\mathbf{G}^i = \frac{\partial}{\partial \mathbf{x}^i}\mathbf{g}(\hat{\mathbf{x}}^{i+}(k))$. Then, for $i \in \mathcal{V}$, the propagation equation (3) can be rewritten as:

$$\hat{\mathbf{x}}^{i-}(k+1) = \mathbf{f}^i(\hat{\mathbf{x}}^{i+}(k), \mathbf{u}^i(k)), \quad (4a)$$
$$\mathbf{P}^{i-}(k+1) = \mathbf{F}^i(k)\mathbf{P}^{i+}(k)\mathbf{F}^i(k)^\top + \mathbf{G}^i(k)\mathbf{Q}^i(k)\mathbf{G}^i(k)^\top, \quad (4b)$$
$$\mathbf{P}^{-}_{ij}(k+1) = \mathbf{F}^i(k)\mathbf{P}^{+}_{ij}(k)\mathbf{F}^j(k)^\top, \quad j \in \mathcal{V}\backslash\{i\}. \quad (4c)$$

Centralized EKF CL Update

While there are no relative measurements in the network, no update happens, therefore,

$$\hat{\mathbf{x}}^{+}(k+1) = \hat{\mathbf{x}}^{-}(k+1), \quad \mathbf{P}^{+}(k+1) = \mathbf{P}^{-}(k+1).$$

We assume that only one relative pose measurement takes place at each time. Let robot a make a relative pose measurement of robot b. The EKF update equation is obtained as follows. The residual of the relative pose measurement and its covariance are, respectively,

$$\mathbf{r}^a = \mathbf{z}_{ab} - \mathbf{h}_{ab}(\hat{\mathbf{x}}^{a-}(k+1), \hat{\mathbf{x}}^{b-}(k+1)), \quad (5a)$$
$$\mathbf{S}_{ab} = \mathbf{H}_{ab}(k+1)\mathbf{P}^{-}(k+1)\mathbf{H}_{ab}(k+1)^\top + \mathbf{R}^a(k+1), \quad (5b)$$

where (without loss of generality we let $a < b$)

$$\mathbf{H}_{ab} = [\overset{1}{\mathbf{0}} \ \overset{\cdots}{\cdots} \ \overset{a}{-\tilde{\mathbf{H}}_a} \ \overset{a+1}{\mathbf{0}} \ \overset{\cdots}{\cdots} \ \overset{b}{\tilde{\mathbf{H}}_b} \ \overset{b+1}{\mathbf{0}} \ \overset{\cdots}{\cdots}],$$
$$\tilde{\mathbf{H}}_a(k+1) = -\frac{\partial}{\partial \mathbf{x}^a}\mathbf{h}_{ab}(\hat{\mathbf{x}}^{a-}(k+1), \hat{\mathbf{x}}^{b-}(k+1)), \quad (6)$$
$$\tilde{\mathbf{H}}_b(k+1) = \frac{\partial}{\partial \mathbf{x}^b}\mathbf{h}_{ab}(\hat{\mathbf{x}}^{a-}(k+1), \hat{\mathbf{x}}^{b-}(k+1)).$$

Substituting for (6) in (5b), we have:

$$\begin{aligned}\mathbf{S}_{ab} = & \ \mathbf{R}^a(k+1) + \tilde{\mathbf{H}}_a(k+1)\mathbf{P}^{a-}(k+1)\tilde{\mathbf{H}}_a(k+1)^\top \\ & + \tilde{\mathbf{H}}_b(k+1)\mathbf{P}^{b-}(k+1)\tilde{\mathbf{H}}_b(k+1)^\top \\ & - \tilde{\mathbf{H}}_b(k+1)\mathbf{P}^{-}_{ba}(k+1)\tilde{\mathbf{H}}_a(k+1)^\top \\ & - \tilde{\mathbf{H}}_a(k+1)\mathbf{P}^{-}_{ab}(k+1)\tilde{\mathbf{H}}_b(k+1)^\top. \end{aligned} \quad (7)$$

Then, the Kalman filter gain is given by

$$\mathbf{K}(k+1) = \mathbf{P}^{-}(k+1)\mathbf{H}_{ab}(k+1)^\top \mathbf{S}_{ab}{}^{-1}.$$

We partition $\mathbf{K}$ as $\mathbf{K} = \left[\mathbf{K}_1^\top, \cdots, \mathbf{K}_N^\top\right]^\top$, where $\mathbf{K}_i \in \mathbb{R}^{n^i \times n_z^i}$ is the portion of the Kalman gain used to update the pose estimate of the robot $i \in \mathcal{V}$. Then, for all $i \in \mathcal{V}$,

$$\mathbf{K}_i = (\mathbf{P}^{-}_{ib}(k+1)\tilde{\mathbf{H}}_b^\top - \mathbf{P}^{-}_{ia}(k+1)\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-1}. \quad (8)$$

Finally, the collective pose update and covariance update equations for the network are:

$$\hat{\mathbf{x}}^{+}(k+1) = \hat{\mathbf{x}}^{-}(k+1) + \mathbf{K}(k+1)\mathbf{r}^a,$$
$$\mathbf{P}^{+}(k+1) = \mathbf{P}^{-}(k+1) - \mathbf{K}(k+1)\mathbf{S}_{ab}\mathbf{K}(k+1)^\top,$$

where for $i \in \mathcal{V}$ and $j \in \mathcal{V}\backslash\{i\}$, it can be expanded as:

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i-}(k+1) + \mathbf{K}_i\mathbf{r}^a, \quad (9a)$$
$$\mathbf{P}^{i+}(k+1) = \mathbf{P}^{i-}(k+1) - \mathbf{K}_i\mathbf{S}_{ab}\mathbf{K}_i^\top, \quad (9b)$$
$$\mathbf{P}^{+}_{ij}(k+1) = \mathbf{P}^{-}_{ij}(k+1) - \mathbf{K}_i\mathbf{S}_{ab}\mathbf{K}_j^\top. \quad (9c)$$

Observe that, despite having decoupled equations of motion, the source of the coupling in the propagation phase is the cross-covariance equation (4c). Upon an incidence of a relative measurement between robots a and b, this term becomes non-zero and its evolution in time requires the information of these two robots. Thus, these two robots have to either communicate with each other all the time or a centralized operation has to take over the propagation stage. As the incidences of relative measurements grow, more non-zero cross-covariance terms are created and the communication cost to perform the propagation grows, requiring the data exchange all the time with either a Fusion Center (FC) or all-to-all robot communications, even when there is no relative measurement in the network. The update equations (9) are also coupled and their calculations need in principle a FC.

IV. THE *Interim Master D-CL* ALGORITHM

In this section, we present our proposed *Interim Master D-CL* algorithm which is a decentralized implementation of the centralized CL algorithm of the previous section. Here, we use the assumption below which generically is valid for mobile robot models:

Assumption 1: $\mathbf{F}^i(k)$ is invertible for all $k \geq 0$ and $i \in \mathcal{V}$.

We start by introducing the new variables we use to develop this decentralized algorithm. Let $\mathbf{\Phi}^i \in \mathbb{R}^{n^i \times n^i}$, for all $i \in \mathcal{V}$, be a time-varying variable that is initialized at $\mathbf{\Phi}^i(0) = \mathbf{I}_{n^i}$, which evolves as:

$$\mathbf{\Phi}^i(k+1) = \mathbf{F}^i(k)\mathbf{\Phi}^i(k).$$

Then, we write the propagated cross-covariances (4c) as:

$$\mathbf{P}^{-}_{ij}(k+1) = \mathbf{\Phi}^i(k+1)\bar{\mathbf{P}}_{ij}(k)\mathbf{\Phi}^j(k+1)^\top, \tag{10}$$

where $\bar{\mathbf{P}}_{ij} \in \mathbb{R}^{n^i \times n^j}$, for $i, j \in \mathcal{V}$ and $i \neq j$, is a time-varying variable that is initialized at $\bar{\mathbf{P}}_{ij}(0) = \mathbf{0}_{n^i \times n^j}$. When there is no relative measurement at time $k+1$, (10) results in $\bar{\mathbf{P}}_{ij}(k+1) = \bar{\mathbf{P}}_{ij}(k)$. Next, when there is a relative measurement, we rewrite the update equations (7) and (8) of the centralized CL algorithm by replacing the cross-covariance terms by (10):

$$\begin{aligned}\mathbf{S}_{ab} = \mathbf{R}^a &+ \tilde{\mathbf{H}}_a\mathbf{P}^{a\text{-}}\tilde{\mathbf{H}}_a^\top + \tilde{\mathbf{H}}_b\mathbf{P}^{b\text{-}}\tilde{\mathbf{H}}_b^\top - \\ &\tilde{\mathbf{H}}_a\mathbf{\Phi}^a(k+1)\bar{\mathbf{P}}_{ab}(k)\mathbf{\Phi}^b(k+1)^\top\tilde{\mathbf{H}}_b^\top - \\ &\tilde{\mathbf{H}}_b\mathbf{\Phi}^b(k+1)\bar{\mathbf{P}}_{ba}(k)\mathbf{\Phi}^a(k+1)^\top\tilde{\mathbf{H}}_a^\top,\end{aligned} \tag{11}$$

and the Kalman gain is

$$\mathbf{K}_i = \mathbf{\Phi}^i(k+1)\mathbf{D}_i, \quad i \in \mathcal{V},$$

where

$$\begin{aligned}&\mathbf{D}_i = (\bar{\mathbf{P}}_{ib}(k)\mathbf{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top - \bar{\mathbf{P}}_{ia}(k)\mathbf{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-1}, \; i \in \mathcal{V}\backslash\{a,b\}\\ &\mathbf{D}_a = (\bar{\mathbf{P}}_{ab}(k)\mathbf{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top - (\mathbf{\Phi}^a)^{-1}\mathbf{P}^{a\text{-}}\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-1},\\ &\mathbf{D}_b = ((\mathbf{\Phi}^b)^{-1}\mathbf{P}^{b\text{-}}\tilde{\mathbf{H}}_b^\top - \bar{\mathbf{P}}_{ba}(k)\mathbf{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-1}.\end{aligned} \tag{12}$$

Notice that due to Assumption 1, $\mathbf{\Phi}^i(k)$, for all $k \geq 0$ and $i \in \mathcal{V}$, is invertible. Let $\bar{\mathbf{r}}^a = (\mathbf{S}_{ab})^{-\frac{1}{2}}\mathbf{r}^a$, and $\bar{\mathbf{D}}_i = \mathbf{D}_i(\mathbf{S}_{ab})^{\frac{1}{2}}$, $i \in \mathcal{V}$. Then, we can write the state estimate and covariance equations (9a) and (9b) as follows:

$$\begin{aligned}&\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i\text{-}}(k+1) + \mathbf{\Phi}^i(k+1)\,\bar{\mathbf{D}}_i\,\bar{\mathbf{r}}^a,\\ &\mathbf{P}^{i+}(k+1) = \mathbf{P}^{i\text{-}}(k+1) - \mathbf{\Phi}^i(k+1)\bar{\mathbf{D}}_i\bar{\mathbf{D}}_i^\top\mathbf{\Phi}^i(k+1)^\top.\end{aligned}$$

For $i \neq j$ and $i, j \in \mathcal{V}$, we let

$$\bar{\mathbf{P}}_{ij}(k+1) = \bar{\mathbf{P}}_{ij}(k) - \bar{\mathbf{D}}_i\bar{\mathbf{D}}_j^\top,$$

then the cross-covariance update (9c) can be rewritten as:

$$\mathbf{P}^{+}_{ij}(k+1) = \mathbf{\Phi}^i(k+1)\bar{\mathbf{P}}_{ij}(k+1)\mathbf{\Phi}^j(k+1)^\top.$$

Therefore, at time $k+2$, the propagated cross-covariances satisfy (10). As such, we can reproduce the effect of the cross-covariance terms of the centralized CL using the variables $\mathbf{\Phi}^i(k)$'s and $\bar{\mathbf{P}}_{ij}$'s. Examining (5a), (6), (11) and (12) shows that robot a can calculate these terms by acquiring $\hat{\mathbf{x}}^{b\text{-}}(k+1) \in \mathbb{R}^{n^b}$, $\mathbf{\Phi}^b(k+1) \in \mathbb{R}^{n^b \times n^b}$, and $\mathbf{P}^{b\text{-}}(k+1) \in \mathbb{M}_{n^b}$ from robot b if it knew $\bar{\mathbf{P}}_{ij}(k)$, $\forall i, j \in \mathcal{V}$. Then robot a can assume the role of the interim master and issue the update terms for other robots in the network. Based on this observation, we develop our *Interim Master D-CL* algorithm by keeping a local copy of $\bar{\mathbf{P}}_{lj}$'s at each robot $i \in \mathcal{V}$, i.e., $\bar{\mathbf{P}}^i_{jl}$ for all $j \in \mathcal{V}\backslash\{N\}$ and $l \in \{j+1, \cdots, N\}$–because of the symmetry of the covariance matrix we only need to save, e.g., the upper triangular part of this matrix. In the following we assume that if $\bar{\mathbf{P}}^i_{jl}$ is not explicitly maintained by robot i, the robot substitutes the value of $(\bar{\mathbf{P}}^i_{lj})^\top$ for it. The *Interim Master D-CL* works as follows:

***Interim Master D-CL* Initialization**

Every robot $i \in \mathcal{V}$ initializes its filter as follows:

$$\begin{aligned}&\hat{\mathbf{x}}^{i+}(0) \in \mathbb{R}^{n^i}, \quad \mathbf{P}^{i+}(0) \in \mathbb{M}_{n^i}, \quad \mathbf{\Phi}^i(0) = \mathbf{I}_{n^i},\\ &\bar{\mathbf{P}}^i_{jl}(0) = \mathbf{0}_{n^l \times n^j}, \; j \in \mathcal{V}\backslash\{N\}, \; l \in \{j+1, \cdots, N\}.\end{aligned} \tag{13}$$

***Interim Master D-CL* Propagation**

Every robot $i \in \mathcal{V}$ propagates the variables below:

$$\begin{aligned}&\hat{\mathbf{x}}^{i\text{-}}(k+1) = \mathbf{f}^i(\hat{\mathbf{x}}^{i+}(k), \mathbf{u}^i(k)), \quad \mathbf{\Phi}^i(k+1) = \mathbf{F}^i(k)\mathbf{\Phi}^i(k),\\ &\mathbf{P}^{i\text{-}}(k+1) = \mathbf{F}^i(k)\mathbf{P}^{i+}(k)\mathbf{F}^i(k)^\top + \mathbf{G}^i(k)\mathbf{Q}^i(k)\mathbf{G}^i(k)^\top.\end{aligned}$$

***Interim Master D-CL* Update**

While there are no relative measurements in the network, every robot $i \in \mathcal{V}$ updates its variables as follows:

$$\begin{aligned}&\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i\text{-}}(k+1), \quad \mathbf{P}^{i+}(k+1) = \mathbf{P}^{i\text{-}}(k+1),\\ &\bar{\mathbf{P}}^i_{jl}(k+1) = \bar{\mathbf{P}}^i_{lj}(k), \quad j \in \mathcal{V}\backslash\{N\}, \; l \in \{j+1, \cdots, N\}.\end{aligned}$$

If there is a robot a that makes a measurement with respect to another robot b, then robot a is declared as the interim master and acquires the following information from robot b:

$$\textit{landmark-message} = \Big(\hat{\mathbf{x}}^{b\text{-}}(k+1), \mathbf{\Phi}^b(k+1), \mathbf{P}^{b\text{-}}(k+1)\Big).$$

Robot a makes the following calculations upon receiving the *landmark-message*:

$$\begin{aligned}\mathbf{r}^a &= \mathbf{z}_{ab} - \mathbf{h}_{ab}(\hat{\mathbf{x}}^{b\text{-}}, \hat{\mathbf{x}}^{a\text{-}}),\\ \mathbf{S}_{ab} &= \mathbf{R}^a + \tilde{\mathbf{H}}_a\mathbf{P}^{a\text{-}}\tilde{\mathbf{H}}_a^\top + \tilde{\mathbf{H}}_b^\top\mathbf{P}^{b\text{-}}\tilde{\mathbf{H}}_b\\ &\quad - \tilde{\mathbf{H}}_a\mathbf{\Phi}^a\bar{\mathbf{P}}^a_{ab}\mathbf{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top - \tilde{\mathbf{H}}_b\mathbf{\Phi}^b\bar{\mathbf{P}}^a_{ba}\mathbf{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top,\\ \bar{\mathbf{D}}_a &= (\mathbf{\Phi}^{a-1}\mathbf{\Phi}^a\bar{\mathbf{P}}^a_{ab}\mathbf{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top - \mathbf{\Phi}^{a-1}\mathbf{P}^{a\text{-}}\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-\frac{1}{2}},\\ \bar{\mathbf{D}}_b &= (\mathbf{\Phi}^{b-1}\mathbf{P}^{b\text{-}}\tilde{\mathbf{H}}_b^\top - \bar{\mathbf{P}}^a_{ba}\mathbf{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-\frac{1}{2}},\end{aligned}$$

where $\tilde{\mathbf{H}}_a(k+1) = \tilde{\mathbf{H}}_a(\hat{\mathbf{x}}^{a-}, \hat{\mathbf{x}}^{b-})$ and $\tilde{\mathbf{H}}_b(k+1) = \tilde{\mathbf{H}}_b(\hat{\mathbf{x}}^{a-}, \hat{\mathbf{x}}^{b-})$ are obtained using (6). We assume that the communication graph has a spanning tree rooted at the interim master, (seeFig. 1). The interim master passes the following data, either directly or indirectly (by message passing), to the rest of the robots in the network:

$$\textit{update-message} = \Big(a, b, \bar{\mathbf{r}}^a, \bar{\mathbf{D}}_a, \bar{\mathbf{D}}_b, \boldsymbol{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top \mathbf{S}_{ab}{}^{-\frac{1}{2}}, \boldsymbol{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top \mathbf{S}_{ab}{}^{-\frac{1}{2}}\Big).$$

Every robot $i \in \mathcal{V}$, upon receiving the *update-message*, first calculates, $\forall j \in \mathcal{V}\backslash\{a, b\}$, using information obtained at k:

$$\bar{\mathbf{D}}_j = \bar{\mathbf{P}}^i_{jb}\boldsymbol{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top \mathbf{S}_{ab}{}^{-\frac{1}{2}} - \bar{\mathbf{P}}^i_{ja}\boldsymbol{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top \mathbf{S}_{ab}{}^{-\frac{1}{2}},$$

and then updates the following variables where $j \in \mathcal{V}\backslash\{N\}, l \in \{j+1, \cdots, N\}$:

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i-}(k+1) + \boldsymbol{\Phi}^i(k+1)\,\bar{\mathbf{D}}_i\,\bar{\mathbf{r}}^a, \quad (14a)$$

$$\mathbf{P}^{i+}(k+1) = \mathbf{P}^{i-}(k+1) - \boldsymbol{\Phi}^i(k+1)\bar{\mathbf{D}}_i\bar{\mathbf{D}}_i^\top\boldsymbol{\Phi}^i(k+1)^\top, \quad (14b)$$

$$\bar{\mathbf{P}}^i_{jl}(k+1) = \bar{\mathbf{P}}^i_{jl}(k) - \bar{\mathbf{D}}_j\bar{\mathbf{D}}_l^\top. \quad (14c)$$

Remark 4.1 (Multiple synchronized relative measurements): To accommodate multiple synchronized relative measurements in the network, we use sequential updating (c.f. [13, ch. 3],[14]). In the Kalman filter development, sequential updating is possible under the assumption that the measurements across *time* and *sensors* are white sequences. To implement a sequential updating procedure in the *Interim Master D-CL* algorithm, we assume that all robots have an identical pre-specified the *sequential-updating-order* guideline indicating the priority order for robots to request the *landmark-message* and broadcast the update-message. One can expect that the updating order should not dramatically change the results. Discussion regarding the update ordering can be found in [14, page 10] and references therein. The sequential updating procedure in the *Interim Master D-CL* algorithm is then as follows: (a) every robot $i \in \mathcal{V}$ making relative measurements informs the entire team that it has made N_z^i relative measurements; (b) in the order dictated by *sequential-updating-order*, the interim master robots, one by one, proceed by requesting the *landmark-message* from their landmarks and (c) broadcasting the *update-message*. □

Relative measurements help the robots improve their localization accuracy but they can not bound the overall uncertainty. As shown in [7], even when all the robots in the team are making relative measurements simultaneously, the observability matrix of the collective system is rank deficient. This rank deficiency can be removed by incorporating absolute pose measurements in the the process. As such, the tracking performance can be improved significantly if robots have occasional absolute positioning information, e.g., via GPS or relative measurements taken from a fixed landmark with a priori known absolute location. The inclusion of absolute measurements in the *Interim Master D-CL* is straightforward. The robot making an absolute measurement is an interim master that can calculate the *update-message* using only its own data and then broadcast it to the team.

Finally, observe that the *Interim Master D-CL* algorithm is robust to permanent robot dropouts from the network. The operation only suffers from a processing cost until all robots become aware of the dropout. Also, notice that an external authority, e.g., a search-and-rescue chief, who needs to obtain the location of any robot, can obtain this location update in any rate (s)he wishes to by communicating with that robot. This reduces the communication cost of the operation.

A. Complexity analysis

For the sake of an objective performance evaluation, we provide a thorough study of the computational complexity, the memory usage, as well as communication cost per robot per time-step of the *Interim Master D-CL* algorithm in terms of the size of the mobile robot team N.

In the *Interim Master D-CL* algorithm, at the propagation stage the computations per robot are independent of the size of the team but at the update stage, for each measurement update, because of (14c), the computation of every robot is of order $N(N-1)/2$. As multiple relative measurements are processed sequentially, the computational cost per robot at the completion of any update stage depends on the number of the relative measurements in the team, henceforth denoted by N_z. Then, the computational cost per robot is $O(N_z \times N^2)$, implying a computational complexity of order $O(N^4)$ for the worst case where all the robots take relative measurement with respect to all the other robots in the team, i.e., $N_z = N(N-1)$. The storage cost per robot is of order $O(N^2)$ which, due to the recursive nature of the *Interim Master D-CL* algorithm, is independent of N_z. This cost is due to the initialization (13) and update equation (14c) which are of order $N(N-1)/2$. We complete our analysis by evaluating the communication cost. There is no communication required in the propagation stage of the *Interim Master D-CL* algorithm. However at the update stage, due to the actions outlined in Remark 4.1 intra-network communications are needed. Recall that every robot re-broadcasts any received message other than their landmark-messages. Let N_r be the number of the robots that have made a relative measurement at the current time. Therefore, to fulfill the steps (a) and (c) of the sequential updating in Remark 4.1, every robot will end up broadcasting, respectively, N_r and N_z times. Every robot can be a master of N_b robots and/or a landmark of N_a robots, requiring that robot to, respectively, broadcast N_b requests and N_a landmark-messages, to fulfill step (b). As $N_a \leq N_r \leq N_z$ and $N_b < N_z$, then the total number of broadcast per robot is of order $O(N_z)$, implying a worst case $(N_z = N(N-1))$ broadcast cost of $O(N^2)$ per robot. If the communication range is unbounded, the broadcast cost per robot is $O(\max\{N_b, N_a\})$, with the worst case cost of order $O(N)$. The communication message size of each robot in both single or multiple relative measurements is independent of the group size N and as such for the worst case scenario the communication message size is of order $O(1)$.

The results of the analysis above are summarized in Table I and are compared to those of a trivial decentralized implementation of the EKF for CL (denoted by T-D-CL) in which every robot $i \in \mathcal{V}$ at the propagation stage computes (4)– using the broadcasted $\mathbf{F}^j(k)$ from every other team member

TABLE I: Complexity analysis per robot of the *Interim Master D-CL* algorithm (denoted by IM-D-CL) compared to that of the trivial decentralized implementation of EKF for CL (denoted by T-D-CL) introduced in Subsection IV-A.

	Computation		Storage		Broadcast*		Message Size		Connectivity	
Algorithm	IM-D-CL	T-D-CL	IM-D-CL	T-D-CL	IM-D-CL	T-D-CL	IM-D-CL	T-D-CL	IM-D-CL	T-D-CL
Propagation	$O(1)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	0	$O(N)$	0	$O(1)$	None	strongly connected digraph
Update per N_z relative measur.	$O(N_z \times N^2)$	$O(N_z \times N^2)$	$O(N^2)$	$O(N^2)$	$O(N_z)$	$O(N_z)$	$O(1)$	$O(1)$	spanning tree rooted at the master robots	
Overall worst case	$O(N^4)$	$O(N^4)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(1)$	$O(1)$		

*Broadcast cost is for multi-hop communication. If the communication range is unbounded, the broadcast cost per robot is $O(\max\{N_b, N_a\})$ with the worst cost of $O(N)$.

$j \in \mathcal{V}\backslash\{i\}$–and at the update stage computes (8) and (9)– using the broadcast (a, b, $\mathbf{r}^a$, $\mathbf{S}_{ab}$, $\tilde{\mathbf{H}}_a$, $\tilde{\mathbf{H}}_b$, $\mathbf{R}^a$, $\mathbf{P}^{a-}$, $\mathbf{P}^{b-}$) from robot a that has made relative measurement from robot b. Robot a calculates $\mathbf{S}_{ab}$, $\tilde{\mathbf{H}}_a$, $\tilde{\mathbf{H}}_b$ by requesting ($\hat{\mathbf{x}}^{b-}$, $\mathbf{P}^{b-}$) from robot b. We assume that multiple measurements are processed sequentially and the communication procedure is multi-hop. Although the overall cost of the T-D-CL algorithm is comparable with the *Interim Master D-CL* algorithm, this implementation has a more stringent communication connectivity condition, requiring a *strongly connected digraph* topology (i.e., all the nodes on the communication graph can be reached by every other node on the graph) at each time-step, regardless of whether there is a relative measurement incidence in the team. As an example, notice that the communication graph of Fig. 1 is not strongly connected and as such the T-D-CL algorithm can not be implemented whereas the *Interim Master D-CL* algorithm can be. Recall that the *Interim Master D-CL* algorithm needs no communication at the propagation stage and it only requires an existence of a spanning tree rooted at the robot making the relative measurement at the update stage. Finally, the *Interim Master D-CL* algorithm incurs less computational cost at the propagation stage.

We close this section with a comparison study with respect to the decentralized MAP algorithm of [8]. The simulations reported in [8] indicate that the MAP strategy for CL is less conservative than the EKF strategy. However, this improvement can come with a demanding computation/communication/storage cost as a result of the MAP strategy's batch processing nature. Recall that the MAP computes the localization estimates for the entire time-steps until the current time-step k_c, as opposed to the Kalman filtering, a recursive algorithm, which computes only the current localization estimate every time-step. The decentralized algorithm of [8] consists of a 7-step procedure, required to be repeated $k_c N$ times, with a reported computational and a broadcast cost of, respectively, $\{O(k_c N), O(N + \log(N)), O(k_c), O(k_c + \log(N)), O(k_c N), O(k_c + \log(N)), O(k_c N)\}$ and $\{0, O(1), 0, O(1), O(k_c), O(1), 0\}$ per repetition regardless of wether there is a relative measurement among the team members. Such procedure results in a total computational complexity of order $O(k_c^2 N^2)$ and a broadcast cost of $O(k_c^2 N)$ per robot. Finally, the storage cost per robot of the D-CL algorithm of [8] is of order $O(k_c N)$. As such as time elapses and k_c grows the cost of D-CL algorithm of [8] can become substantially larger than that of the *Interim Master D-CL* algorithm. Note that the broadcast cost of [8] is calculated based on the assumption that a broadcast by any robot can reach the entire team. This cost will go up to deal with multi-hop communications by means of re-broadcasting.

V. Conclusions

For a team of communicating robots, we presented a decentralized cooperative localization algorithm that is exactly equivalent to the centralized EKF algorithm of [7]. In this decentralized algorithm, the propagation stage is fully decoupled i.e., the propagation is a local calculation and no intra-network communication is needed. The communication between robots is only required in the update stage when one robot makes a relative measurement with respect to another robot. The algorithm declares the robot made the measurement as interim master that can, by using the data acquired from the landmark robot, calculate the update terms for the rest of the team and deliver it to them by broadcast. Future work involves extension of the algorithm to let new robots join the group and also study the effect of missed broadcast messages as well as asynchronous operation.

References

[1] J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 376–382, June 1991.

[2] G. Dissanayake, P. Newman, H. F. Durrant-Whyte, , S. Clark, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001. Check.

[3] S. Roumeliotis, *Robust mobile robot localization: from single-robot uncertainties to multi-robot interdependencies*. PhD thesis, University of Southern California, 2000.

[4] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem," in *Int. Conference on Distributed Autonomous Robotic Systems*, (Fukuoka, Japan), pp. 299–308, June 2002.

[5] A. Bahr, M. R. Walter, and J. Leonard, "Consistent cooperative localization," in *IEEE Int. Conf. on Robotics and Automation*, (Kobe, Japan), pp. 8908–8913, May 2009.

[6] 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), pp. 4394–4400, May 2009.

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

[8] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," in *IEEE Int. Conf. on Robotics and Automation*, (Kobe, Japan), pp. 1402–1409, May 2009.

[9] K. Y. K. Leung, T. D. Barfoot, and H. H. T. Liu, "Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 62–77, 2010.

[10] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, “Nonparametric belief propagation for self-localization of sensor networks,” *IEEE Journal of Selected Areas in Communications*, vol. 23, no. 4, pp. 809–819, 2005.

[11] J. Nilsson, D. Zachariah, I. Skog, and P. Händel, “Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging,” *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 164, 2013.

[12] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Applied Mathematics Series, Princeton University Press, 2009. Available at http://www.coordinationbook.info.

[13] C. Leondes, ed., *Advances in Control Systems Theory and Application*, vol. 3. New York: Academic Press, 1966.

[14] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion, a Handbook of Algorithms*. Storts, CT, USA: YBS Publishing, 2011.