

UC Irvine

UC Irvine Previously Published Works

Title

Cooperative Localization for Mobile Agents A RECURSIVE DECENTRALIZED ALGORITHM BASED ON KALMAN-FILTER DECOUPLING

Permalink

<https://escholarship.org/uc/item/1bm5w61r>

Journal

IEEE CONTROL SYSTEMS MAGAZINE, 36(2)

ISSN

1066-033X

Authors

Kia, Solmaz S
Rounds, Stephen
Martinez, Sonia

Publication Date

2016

DOI

10.1109/MCS.2015.2512033

Peer reviewed

Cooperative Localization for Mobile Agents

A recursive decentralized algorithm based on

Kalman filter decoupling

Solmaz S. Kia, Stephen Rounds and Sonia Martínez

Technological advances in ad-hoc networking and in miniaturization of electro-mechanical systems have enabled us to use large numbers of mobile agents (e.g., mobile robots, human agents, unmanned underwater vehicles or spacecraft) to perform surveillance, search and rescue, transport and delivery tasks—beyond the capabilities of a single device—in aerial, underwater, space, and land environments. However, the successful execution of higher level tasks undertaken by these mobile agents often hinges on accurate position information, which is needed in lower level locomotion and path planning algorithms for agent control. Common techniques for localization in mobile agent applications are the classical pre-installed beacon-based localization algorithms [1], fixed feature-based Simultaneous Localization and Mapping

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

(SLAM) algorithms [2] and GPS navigation [3], for further details see Fig. 1.

Despite the availability of this variety of localization techniques for various environment types, there are still mobile agent team operations, for example, in search and rescue [4], [5], environmental monitoring [6], [7], and oceanic exploration [8], for which the environment is often uncharted, is not accessible in advance, does not have distinct features or the features are dynamic or not revisited during the operation. Moreover in such applications the environment is fully or partially GPS denied. Cooperative localization (CL) seems to be the best localization technique for such applications. In CL a group of mobile agents, with processing and communication capabilities, use relative measurements with respect to each other (no reliance on external features) as a feedback signal to *jointly estimate* the poses of the team members, resulting in an increased accuracy for the entire team. CL is particularly appealing, because it can expand the benefit of intermittent accurate absolute localization information access of some team members to the rest of the group through the coupling that is created in state estimations across team members. Another nice feature of the CL strategy is its cost-effectiveness, as it does not require extra hardware beyond the operational components normally used in aforementioned cooperative tasks. In such tasks, agents are normally equipped with unique identifiers and sensors which enable them to locate other members. And, to coordinate, these agents often broadcast status information to one another. In addition, given the wide and affordable availability of communication devices, CL has also emerged as an augmentation system compensation system for poor odometric measurements, noisy and distorted measurements from other sensor suites such as IMU systems on board of mobile vehicles, see e.g., [9].

Cooperative Localization

Consider a team of N mobile agents with communication, processing and measurement capabilities. Every agent has a bounded communication range. Communications can happen either in a single broadcast to the entire team or in multi-hop fashion, i.e., every agent re-broadcasts every received message intended to reach the entire team, see Fig. 2. Each agent has a detectable unique identifier (UID) which, without loss of generality, here we assume to be a unique integer belonging to the set $\mathcal{V} = \{1, \dots, N\}$. Using a set of so-called proprioceptive sensors every agent $i \in \mathcal{V}$ measures its self-motion and uses it to propagate its equations of motion

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

where $\mathbf{x}^i \in \mathbb{R}^{n^i}$, $\mathbf{u}^i \in \mathbb{R}^{m^i}$, and $\mathbf{n}^i \in \mathbb{R}^{p^i}$ are, respectively, the pose vector, the input vector and the process noise vector of agent i . Here, $\mathbf{f}^i(\mathbf{x}^i, \mathbf{u}^i)$ and $\mathbf{g}^i(\mathbf{x}^i)$, are, respectively, the system function and process noise coefficient function of the agent $i \in \mathcal{V}$. The team can consist of heterogeneous agents, nevertheless, the collective motion equation of the team can be represented by

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

where, $\mathbf{f}(\mathbf{x}, \mathbf{u}) = (\mathbf{f}^1(\mathbf{x}^1, \mathbf{u}^1), \dots, \mathbf{f}^N(\mathbf{x}^N, \mathbf{u}^N))$ and $\mathbf{g}(\mathbf{x}) = \text{Diag}(\mathbf{g}^1(\mathbf{x}^1), \dots, \mathbf{g}^N(\mathbf{x}^N))$.

One can clearly see that if agents only rely on propagating their equations of motion in (1) using self-motion measurements, because of the noise terms $\mathbf{n}^i(k)$, this pose estimate will diverge. To bound this pose estimation error, a CL strategy can be employed. Thus, let every agent $i \in \mathcal{V}$ also carry exteroceptive sensors to monitor the environment to detect, uniquely, the

other agents $j \in \mathcal{V}$ in the team and take relative measurements

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

where $\mathbf{z}_{ij} \in \mathbb{R}^{n_z^i}$ from them, e.g., range or bearing measurements, or both. Here, $\mathbf{h}_{ij}(\mathbf{x}^i, \mathbf{x}^j)$ is the measurement model and $\boldsymbol{\nu}^i$ is the measurement noise of agent $i \in \mathcal{V}$. To demonstrate the features of a CL strategy and discuss challenges in its design and implementation, let us employ an Extended Kalman Filter (EKF) to the collective system model of the team of mobile agents described above, following [10]. To do this, we assume that the process noises \mathbf{n}^i and the measurement noise $\boldsymbol{\nu}^i$, $i \in \mathcal{V}$, are independent zero-mean white Gaussian processes with, respectively, a known positive definite variance $\mathbf{Q}^i(k) = E[\mathbf{n}^i(k)^\top \mathbf{n}^i(k)]$ and $\mathbf{R}^i(k) = E[\boldsymbol{\nu}^i(k)^\top \boldsymbol{\nu}^i(k)]$. All sensor noises are assumed to be white and mutually uncorrelated. Here, we assume that all sensor measurements are synchronized. Then, the application of an EKF over the collective motion model (1) using the relative measurement model (2) results in the following propagation and update stages. The propagation stage of this algorithm is

$$\hat{\mathbf{x}}^-(k+1) = \mathbf{f}(\hat{\mathbf{x}}^+(k), \mathbf{u}(k)), \quad (3a)$$

$$\mathbf{P}^-(k+1) = \mathbf{F}(k)\mathbf{P}^+(k)\mathbf{F}(k)^\top + \mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}(k)^\top. \quad (3b)$$

where $\mathbf{F} = \text{Diag}(\mathbf{F}^1, \dots, \mathbf{F}^N)$, $\mathbf{G} = \text{Diag}(\mathbf{G}^1, \dots, \mathbf{G}^N)$ and $\mathbf{Q} = \text{Diag}(\mathbf{Q}^1, \dots, \mathbf{Q}^N)$, 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 $\mathbf{G}^i = \frac{\partial}{\partial \mathbf{x}^i} \mathbf{g}(\hat{\mathbf{x}}^{i+}(k))$.

If there exists a relative measurement in the network at some given time $k+1$, the states are updated as follows. The residual of the relative measurement and its covariance are, respectively,

$$\mathbf{r}^a = \mathbf{z}_{ab} - \mathbf{h}_{ab}(\hat{\mathbf{x}}^{a-}(k+1), \hat{\mathbf{x}}^{b-}(k+1)), \quad (4a)$$

$$\mathbf{S}_{ab} = \mathbf{H}_{ab}(k+1)\mathbf{P}^-(k+1)\mathbf{H}_{ab}(k+1)^\top + \mathbf{R}^a(k+1). \quad (4b)$$

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

$$\begin{aligned} \mathbf{H}_{ab}(k) &= \begin{bmatrix} 1 & \cdots & -\tilde{\mathbf{H}}_a^a(k) & \mathbf{0}^{a+1} & \cdots & \tilde{\mathbf{H}}_b^b(k) & \mathbf{0}^{b+1} & \cdots \end{bmatrix}, \\ \tilde{\mathbf{H}}_a^a(k) &= -\frac{\partial}{\partial \mathbf{x}^a} \mathbf{h}_{ab}(\hat{\mathbf{x}}^{a-}(k), \hat{\mathbf{x}}^{b-}(k)), \\ \tilde{\mathbf{H}}_b^b(k) &= \frac{\partial}{\partial \mathbf{x}^b} \mathbf{h}_{ab}(\hat{\mathbf{x}}^{a-}(k), \hat{\mathbf{x}}^{b-}(k)). \end{aligned} \quad (5)$$

Then, the Kalman filter gain is given by

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

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

$$\hat{\mathbf{x}}^+(k+1) = \hat{\mathbf{x}}^-(k+1) + \mathbf{K}(k+1)\mathbf{r}^a, \quad (6a)$$

$$\mathbf{P}^+(k+1) = \mathbf{P}^-(k+1) - \mathbf{K}(k+1)\mathbf{S}_{ab}\mathbf{K}(k+1)^\top. \quad (6b)$$

Because $\mathbf{K}(k+1)\mathbf{S}_{ab}\mathbf{K}(k+1)^\top$ is a positive semi-definite term, the update equation (6b) clearly shows that any relative measurement update results into a reduction of the estimation uncertainty.

To explore the interaction among team members' estimation equations, we express the aforementioned collective form of the EKF CL in terms of its agent-wise components, as shown in Algorithm 1. Here, the Kalman filter gain is partitioned into $\mathbf{K} = \left[\mathbf{K}_1^\top, \dots, \mathbf{K}_N^\top \right]^\top$, 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 agent

$i \in \mathcal{V}$. To process multiple synchronized measurements, *sequential updating* (c.f. e.g., [11, Ch. 3],[12]) is employed.

Algorithm 1 clearly showcases the role of past correlations in a CL strategy. First, observe that, despite having decoupled equations of motion, the source of the coupling in the propagation phase is the cross-covariance equation (14c). Upon an incidence of a relative measurement between agents a and b , this term becomes non-zero and its evolution in time requires the information of these two agents. Thus, these two agents 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. As a result, 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 agent communications, even when there is no relative measurement in the network. The update equations (16) are also coupled and their calculations need in principle a FC. The next observation regarding the role of the cross-covariance terms can be deduced from studying Kalman gain equation (17). As this equation shows, when an agent a takes a relative measurement from agent b , any agent whose pose estimation is correlated with either of agents a and b in the past, (i.e., $\mathbf{P}_{ib}^-(k+1)$ and/or $\mathbf{P}_{ia}^-(k+1)$ are non-zero) has non-zero Kalman gain and as a result the agent benefits from this measurement update.

The following simulation study demonstrates the significance of maintaining an accurate account of cross-covariance terms between the state estimates of the team members. Figure 3 demonstrates the the x -coordinate estimation error (solid line) and 3σ error bound (dashed lines)

of 3 robots moving on a flat terrain when they (a) only propagate their equations of motion using self-motion measurements (black plots), (b) employ an EKF CL ignoring past correlations between the estimations of the robots (blue plots), (c) employ an EKF CL with accurate account of past correlations (red plots). As this figure shows, employing a CL strategy improves the localization error, but, as plots in blue show, ignoring the past correlations (here cross-covariances) among the robots state estimates results in estimations with almost vanished 3σ error bound, an indication of inconsistent estimation. In contrast, by taking the past correlations into account (see red plots), one sees a more consistent estimation, which clearly showcases the benefit of using the relative measurement updates to reduce the estimation error and uncertainty, and to expand this benefit to other robots that are the subject of a relative or an absolute measurement in a given time interval.

Decentralized Cooperative Localization

Based on the observations that

- (a) past correlations cannot be ignored,
- (b) they are useful to increase the localization accuracy of the team,
- (c) the coupling that the correlations create in the state estimation of team members is the main challenge in developing a decentralized cooperative localization algorithm,

one can observe, regardless of the technique, two distinct trends in the design methodology of decentralized cooperative localization algorithms in the literature that we term as “loosely coupled” and “tightly coupled” decentralized cooperative localization (D-CL) strategies (see

Fig. 4).

In loosely coupled D-CL methodology, only one or both of the agents involved in a relative measurement, update their estimates using that measurement. Here, we do not maintain the exact account of the ‘network’ of correlations (see Fig. 4) due to the past relative measurement updates however, to ensure the estimation consistency, we take steps to account for the past correlations. Examples of loosely coupled D-CL are given in [8] and [13]. In the algorithm of [8], only the agent obtaining the relative measurement updates its state. Here, in order to produce consistent estimates, a bank of extended Kalman filters (EKFs) is maintained at each agent. Using an accurate book-keeping of the identity of the agents 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. Although this technique does not impose a particular communication graph on the network, the computational complexity, the large memory demand, and the growing size of information needed at each update time are its main drawbacks. In the approach [13] it is assumed that the relative measurements are in the form of relative pose. This enables the agent taking the relative measurement to use its current pose estimation and the current relative pose measurement to obtain and broadcast a pose and the associated covariance estimation of its landmark agent, the agent the relative measurement is taken from it. Then, the landmark agent uses the covariance intersection method to fuse the newly acquired pose estimation with its own current estimation to increase its estimation accuracy. Another example of use of covariance intersection method for D-CL is given in [14] for the localization of a group of space vehicles communicating over a fixed ring topology. Here, each vehicle propagates a model of the equation of motion of the entire team and at the time of relative pose measurements

fuses its estimation of the collective team model and its landmark vehicle using a covariance intersection method. Even though the covariance intersection method can produce consistent estimations for a loosely coupled D-CL strategy, this method is known to produce conservative estimates. As showcased by the example strategies above, the loosely coupled algorithms have the advantage of not imposing any particular connectivity condition on the team. However, they are conservative by nature, as they do not enable other agent in the network to benefit from the update. The reader interested on technical details of Covariance Intersection can find a brief literature guide in “Covariance Intersection.”

In the tightly coupled D-CL methodology, the goal is to exploit the ‘network’ of correlations created across the team (see Fig. 4), so that the benefit of the update can be extended beyond the agents involved in a given relative measurement. However, this advantage comes at a potentially higher computational, storage and/or communication cost. The dominant trend in developing decentralized cooperative localization algorithms in this way is to distribute the computation of components of a centralized algorithm among team members. Some of the examples for this class of D-CL is given in [15], [10], [16], [17]. In a straightforward fashion, decentralization can be conducted as a multi-centralized CL, wherein each agent broadcasts its own information to the entire team. Then, every agent can calculate and reproduce the centralized pose estimates acting as a fusion center [15]. Besides a high-processing cost for each agent, this scheme requires all-to-all agent communication at the time of each information exchange. A D-CL algorithm distributing computations of an EKF centralized CL algorithm is proposed in [10]. To decentralize the cross-covariance propagation, [10] uses a singular-value decomposition to split each cross-covariance term between the corresponding two agents. Then, each agent propagates

its portion. However, at update times, the separated parts must be combined, requiring an all-to-all agent communication in the correction step. Subsequently, [16] presents a maximum-a-posteriori (MAP) D-CL algorithm in which all the agents in the team calculate parts of the centralized CL. The aforementioned techniques all assume that communication messages are delivered, as prescribed, perfectly all the time. A D-CL approach equivalent to a centralized CL, when possible, which handles both limited communication ranges and time-varying communication graphs is proposed in [17]. This technique uses an information transfer scheme wherein each agent broadcasts all its locally available information to every agent within its communication radius at each time-step. The broadcasted information of each agent includes the past and present measurements, as well as past measurements previously received from other agents. The main drawback of this method is its high communication and storage cost, which may not be affordable in applications with limited communication bandwidth and storage resources. CL techniques to handle system and measurement models with non-Gaussian noises are discussed in [18], [19] but they do not address the communication message dropouts.

In the remainder of this note, we review a recursive D-CL algorithm called *Interim Master D-CL*, proposed in [20], which is exactly equivalent to a centralized EKF for CL, i.e., it is a tightly coupled CL strategy. This 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 agent making the relative measurement as the *interim master*, which provides the rest of the agents with the information they need to update their pose and covariance in a manner that exactly matches those of a centralized EKF for CL. To calculate the update equations, the interim master only

requires information from the *interim landmark*, the agent 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 interaction between agents can be time-varying with the only requirement that the message from the masters must reach every agent in the network. Furthermore, we show that the *Interim Master D-CL* can easily incorporate absolute measurements, and is robust to permanent agent drop-outs. We discuss the storage, computation and communication cost per agent of this algorithm and show that the size of the associated messages is independent of the size of the team.

The *Interim Master D-CL* algorithm

The *Interim Master D-CL* algorithm, which is a decentralized implementation of the centralized CL algorithm, is constructed based on the following observation. Let the last measurement update to be in time-step k and for the m consecutive steps no relative measurement incidence takes place among the team members, i.e., no intermediate measurement update is conducted in this time interval. In such a scenario, the propagated cross-covariance terms for these m consecutive steps are given by

$$\mathbf{P}_{ij}^-(k+l) = \mathbf{F}^i(k+l-1) \cdots \mathbf{F}^i(k) \mathbf{P}_{ij}^+(k) \mathbf{F}^j(k)^\top \cdots \mathbf{F}^j(k+l-1)^\top, \quad l \in \{1, \dots, m\}, \quad (7)$$

for $i \in \mathcal{V}$ and $j \in \mathcal{V} \setminus \{i\}$. That is, at each time-step after k , the propagated cross-covariance term is obtained by recursively multiplying its value at past time-step from left by the Jacobian of the system function of agent i and from the right by the transpose of the Jacobian of the system

function of agent j at that time-step. Based on this observation, Roumeliotis and Bekey in [10] proposed to decompose the last updated cross-covariance term $\mathbf{P}_{ij}^+(k)$ between any agent i with any agent j of the team into two parts (for example using the singular value decomposition technique). Then, agent i will be responsible for propagating the left portion while agent j propagates the right portion. Note that, as long as there is no relative measurement among team members, each agent can propagate its portion of the cross-covariance term locally without a need of communication with others. This was an important result, which lead to a fully decentralized estimation algorithm during the propagation cycle. However, in the update stage, all the agents needed to communicate with one and other to put together the split cross-covariance terms and proceed with the update stage. The approach to obtain *Interim Master D-CL*, which is outlined below, is also based on the special pattern that the cross-covariance propagation equations have in (7). That is, we also remove the explicit calculation of the propagated cross-covariance terms by decomposing them to the intermediate variables that can be propagated by agents locally. However, this alternative decomposition allows every agent to update its pose estimate and its associated covariance in a centralized equivalent manner, using merely an scalable communication message that it receives from the team member that has taken a relative measurement. As such, the *Interim Master D-CL* algorithm removes the necessity of an all-to-all communication in the update stage and replaces it with receiving communication message from a team member that holds the crucial piece of information in the update stage.

In particular, we observe that $\mathbf{P}_{ij}^-(k+l-1)$ in (7) is composed of the following 3 parts: a) $\mathbf{F}^i(k+l-1) \cdots \mathbf{F}^i(k)$ which is local to agent i , b) the $\mathbf{P}_{ij}^+(k)$ that does not change unless there is relative measurement among the team members, and c) $\mathbf{F}^j(k)^\top \cdots \mathbf{F}^j(k+l-1)^\top$ which is

local to agent j . Given this observation, we write the propagated cross-covariances (14c) as:

$$\mathbf{P}_{ij}^-(k+1) = \Phi^i(k+1)\bar{\mathbf{P}}_{ij}(k)\Phi^j(k+1)^\top, \quad (8)$$

where $\Phi^i \in \mathbb{R}^{n^i \times n^i}$, for all $i \in \mathcal{V}$, is a time-varying variable that is initialized at $\Phi^i(0) = \mathbf{I}_{n^i}$ and which evolves as:

$$\Phi^i(k+1) = \mathbf{F}^i(k)\Phi^i(k), \quad (9)$$

and $\bar{\mathbf{P}}_{ij} \in \mathbb{R}^{n^i \times n^j}$, for $i, j \in \mathcal{V}$ and $i \neq j$, which is also 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 $k+1$, (8) results into $\bar{\mathbf{P}}_{ij}(k+1) = \bar{\mathbf{P}}_{ij}(k)$. However, when there is a relative measurement among the team members $\bar{\mathbf{P}}_{ij}$ must be updated. Given this decomposition, as is shown below, we decentralize the propagation cycle of the EKF for CL by requiring that every agent $i \in \mathcal{V}$ to keep a local copy of $\bar{\mathbf{P}}_{lj}$'s of the entire team, i.e., $\bar{\mathbf{P}}_{lj}^i$ for all $j \in \mathcal{V} \setminus \{i\}$ and $l \in \{1, \dots, N\} \setminus \{j\}$.

Next, we derive an expression for $\bar{\mathbf{P}}_{ij}(k+1)$ when there is a relative measurement among team members at time $k+1$, such that at time $k+2$ one can write $\mathbf{P}_{ij}^-(k+2) = \Phi^i(k+2)\bar{\mathbf{P}}_{ij}(k+1)\Phi^j(k+2)^\top$. For this, notice that the update equations (15) and (17) of the centralized CL algorithm can be rewritten by replacing the cross-covariance terms by (8):

$$\begin{aligned} \mathbf{S}_{ab} &= \mathbf{R}^a + \tilde{\mathbf{H}}_a \mathbf{P}^{a-}(k+1) \tilde{\mathbf{H}}_a^\top + \tilde{\mathbf{H}}_b \mathbf{P}^{b-}(k+1) \tilde{\mathbf{H}}_b^\top - \\ &\quad \tilde{\mathbf{H}}_a \Phi^a(k+1) \bar{\mathbf{P}}_{ab}(k) \Phi^b(k+1)^\top \tilde{\mathbf{H}}_b^\top - \tilde{\mathbf{H}}_b \Phi^b(k+1) \bar{\mathbf{P}}_{ba}(k) \Phi^a(k+1)^\top \tilde{\mathbf{H}}_a^\top, \end{aligned} \quad (10)$$

and the Kalman gain is

$$\mathbf{K}_i = \Phi^i(k+1) \bar{\mathbf{D}}_i (\mathbf{S}_{ab})^{-\frac{1}{2}}, \quad i \in \mathcal{V},$$

where

$$\bar{\mathbf{D}}_i = (\bar{\mathbf{P}}_{ib}(k) \Phi^b \tilde{\mathbf{H}}_b^\top - \bar{\mathbf{P}}_{ia}(k) \Phi^a \tilde{\mathbf{H}}_a^\top) \mathbf{S}_{ab}^{-\frac{1}{2}}, \quad i \in \mathcal{V} \setminus \{a, b\}, \quad (11a)$$

$$\bar{\mathbf{D}}_a = (\bar{\mathbf{P}}_{ab}(k) \Phi^b \tilde{\mathbf{H}}_b^\top - (\Phi^a)^{-1} \mathbf{P}^a \tilde{\mathbf{H}}_a^\top) \mathbf{S}_{ab}^{-\frac{1}{2}}, \quad (11b)$$

$$\bar{\mathbf{D}}_b = ((\Phi^b)^{-1} \mathbf{P}^b \tilde{\mathbf{H}}_b^\top - \bar{\mathbf{P}}_{ba}(k) \Phi^a \tilde{\mathbf{H}}_a^\top) \mathbf{S}_{ab}^{-\frac{1}{2}}. \quad (11c)$$

Here, we employ the assumption below which generically is valid for mobile agent models:

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

Notice that due to Assumption 1, $\Phi^i(k)$, for all $k \geq 0$ and $i \in \mathcal{V}$, is invertible. Next, for $i \neq j$ and $i, j \in \mathcal{V}$, we let

$$\bar{\mathbf{P}}_{ij}(k+1) = \bar{\mathbf{P}}_{ij}(k) - \bar{\mathbf{D}}_i \bar{\mathbf{D}}_j^\top. \quad (12)$$

Then, the cross-covariance update (16c) can be rewritten as:

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

Therefore, at time $k+2$, the propagated cross-covariances satisfy (8) where k is replaced by $k+1$. As such, we can reproduce the effect of the cross-covariance terms of the centralized CL using the variables $\Phi^i(k)$'s and $\bar{\mathbf{P}}_{ij}$'s. Notice that we can write the updated state estimate and covariance matrix in the new variables as follows, for $i \in \mathcal{V}$,

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i-}(k+1) + \Phi^i(k+1) \bar{\mathbf{D}}_i \bar{\mathbf{r}}^a, \quad (13)$$

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

where $\bar{\mathbf{r}}^a = (\mathbf{S}_{ab})^{-\frac{1}{2}} \mathbf{r}^a$.

Given the decomposition above, examining (5), (4a), (10) and (11) shows that agent a can calculate these terms by acquiring $\hat{\mathbf{x}}^{b^-}(k+1) \in \mathbb{R}^{n^b}$, $\Phi^b(k+1) \in \mathbb{R}^{n^b \times n^b}$, and $\mathbf{P}^{b^-}(k+1) \in \mathbb{M}_{n^b}$ from agent b if it knew $\bar{\mathbf{P}}_{ij}(k)$, $\forall i, j \in \mathcal{V}$. Then agent a can assume the role of the interim master and issue the update terms for other agents in the network. Based on this observation, we develop the *Interim Master D-CL* algorithm by keeping a local copy of $\bar{\mathbf{P}}_{ij}$'s at each agent $i \in \mathcal{V}$, i.e., $\bar{\mathbf{P}}_{jl}^i$ for all $j \in \mathcal{V} \setminus \{N\}$ and $l \in \{j+1, \dots, N\}$ —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 $\bar{\mathbf{P}}_{jl}^i$ is not explicitly maintained by agent i , the agent substitutes the value of $(\bar{\mathbf{P}}_{lj}^i)^\top$ for it. The *Interim Master D-CL* works as presented in Algorithm 2. This algorithm works based under the following assumption

Assumption 2: We assume that the message from the agent taking the relative measurement, the interim master, can reach the entire team (see Fig. 2).

Remark 0.1 (Multiple synchronized relative measurements): To accommodate multiple synchronized relative measurements in the network, we use sequential updating (c.f. [11, ch. 3],[12]). 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 agents have an identical pre-specified the *sequential-updating-order* guideline indicating the priority order for agents to request the *landmark-message* and broadcast the update-message. It is reasonable to expect that the updating order should not dramatically change the results. Discussion regarding the update ordering can be found in [12, page 10] and references therein. The sequential updating

procedure in the *Interim Master D-CL* algorithm is then as follows: (a) every agent $i \in \mathcal{V}$ 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 agents, one by one, proceed by requesting the *landmark-message* from their landmarks and (c) broadcasting the *update-message*.
□

Relative measurements help the agents improve their localization accuracy but they can not bound the overall uncertainty. As shown in [10], even when all the agents in the team are taking 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 process. As such, the tracking performance can be improved significantly if agents 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 agent 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 agent dropouts from the network. The operation only suffers from a processing cost until all agents 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 agent, can obtain this location update in any rate (s)he wishes to by communicating with that agent. This reduces the communication cost of the operation.

Complexity analysis

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

In the *Interim Master D-CL* algorithm, at the propagation stage the computations per agent are independent of the size of the team but at the update stage, for each measurement update, because of (19c), the computation of every agent is of order $N(N - 1)/2$. As multiple relative measurements are processed sequentially, the computational cost per agent 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 agent is $O(N_z \times N^2)$, implying a computational complexity of order $O(N^4)$ for the worst case where all the agents take relative measurement with respect to all the other agents in the team, i.e., $N_z = N(N - 1)$. The storage cost per agent 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 caused by the initialization (18) and update equation (19c), which are of order $N(N - 1)/2$. We complete the 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 0.1, intra-network communications are needed. Recall that every agent re-broadcasts any received message other than their landmark-messages. Let N_r be the number of the agents that have made a relative measurement at the current time. Therefore, to fulfill the steps (a) and (c) of the sequential

updating in Remark 0.1, every agent will end up broadcasting, respectively, N_r and N_z times. Every agent can be a master of N_b agents and/or a landmark of N_a agents, requiring that agent 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 agent is of order $O(N_z)$, implying a worst case ($N_z = N(N-1)$) broadcast cost of $O(N^2)$ per agent. If the communication range is unbounded, the broadcast cost per agent is $O(\max\{N_b, N_a\})$, with the worst case cost of order $O(N)$. The communication message size of each agent 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 agent $i \in \mathcal{V}$ at the propagation stage computes (14)–using the broadcasted $\mathbf{F}^j(k)$ from every other team member $j \in \mathcal{V} \setminus \{i\}$ –and at the update stage computes (17) and (16)–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 agent a that has made relative measurement from agent b . Agent a calculates $\mathbf{S}_{ab}, \tilde{\mathbf{H}}_a, \tilde{\mathbf{H}}_b$ by requesting $(\hat{\mathbf{x}}^{b-}, \mathbf{P}^{b-})$ from agent 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. 2 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 agent making the relative measurement at the update stage. Finally, the *Interim Master D-CL* algorithm incurs less computational cost at the propagation stage.

Conclusions

In this note, we reviewed the cooperative localization strategy for increasing the localization accuracy of team of communicating mobile agents. This strategy relies on use of agent-to-agent relative measurements (no reliance on external features) as a feedback signal to *jointly estimate* the poses of the team members, resulting in an increased accuracy for the entire team. In particular, we discussed the challenges involved in designing decentralized cooperative localization algorithms. Moreover, we presented a decentralized cooperative localization algorithm that is exactly equivalent to the centralized EKF algorithm of [10]. 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 agents is only required in the update stage when one agent makes a relative measurement with respect to another agent. The algorithm declares the agent made the measurement as interim master that can, by using the data acquired from the landmark agent, calculate the update terms for the rest of the team and deliver it to them by broadcast.

References

- [1] J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):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*, 17(3):229–241, 2001. Check.
- [3] S. Cooper and H. Durrant-Whyte. A kalman filter model for gps navigation of land vehicles. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 157–163, Munich, Germany, 1994.
- [4] J. S. Jennings, G. Whelan, and W. F. Evans. Cooperative search and rescue with a team of mobile robots. In *Int. Conf. on Advanced Robotics*, pages 193–200, 1997.
- [5] A. Davids. Urban search and rescue robots: from tragedy to technology. *IEEE Intelligent Systems*, 17(2):81–83, March-April 2002.
- [6] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, 2000.
- [7] M. Trincavelli, M. Reggente, S. Coradeschi, A. Loutfi, H. Ishida, and A. J. Lilienthal. Towards environmental monitoring with mobile robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 2210–2215, 2008.
- [8] A. Bahr, M. R. Walter, and J. J. Leonard. Consistent cooperative localization. In *IEEE Int. Conf. on Robotics and Automation*, pages 8908–8913, Kobe, Japan, May 2009.
- [9] H. Mokhtarzadeh and D. Gebre-Egziabher. Cooperative inertial navigation. *Navigaton:*

- Journal of the Institute of Navigation*, 61(2):77–94, 2014.
- [10] S. I. Roumeliotis and G. A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.
- [11] C. T. Leondes, editor. *Advances in Control Systems Theory and Application*, volume 3. Academic Press, New York, 1966.
- [12] Y. Bar-Shalom, P. K. Willett, and X. Tian. *Tracking and Data Fusion, a Handbook of Algorithms*. YBS Publishing, Storrs, CT, USA, 2011.
- [13] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis. Decentralized multi-robot cooperative localization using covariance intersection. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 1412–1417, Tokyo, Japan, 2013.
- [14] P. O. Arambel, C. Rago, and R. K. Mehra. Covariance intersection algorithm for distributed spacecraft state estimation. In *American Control Conference*, pages 4398–4403, Arlington, VA, 2001.
- [15] N. Trawny, S. I. Roumeliotis, and G. B. Giannakis. Cooperative multi-robot localization under communication constraints. In *IEEE Int. Conf. on Robotics and Automation*, pages 4394–4400, Kobe, Japan, May 2009.
- [16] 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*, pages 1402–1409, Kobe, Japan, May 2009.
- [17] 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*, 26(1):62–77, 2010.

- [18] 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*, 23(4):809–819, 2005.
- [19] 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*, 2013(164), 2013.
- [20] S.S. Kia, S. Rounds, and S. Martínez. A centralized-equivalent decentralized implementation of extended Kalman filters for cooperative localization. In *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, Chicago, IL, September 2014. To appear.

Algorithm 1 EKF CL (centralized)

Require: Initialization ($k = 0$): For $i \in \mathcal{V}$, the algorithm is initialized at

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

Iteration k

1: Propagation: for $i \in \mathcal{V}$, the propagation equations are:

$$\hat{\mathbf{x}}^{i-}(k+1) = \mathbf{f}^i(\hat{\mathbf{x}}^{i+}(k), \mathbf{u}^i(k)), \quad (14a)$$

$$\mathbf{P}^{i-}(k+1) = \mathbf{F}^i(k)\mathbf{P}^{i+}(k)\mathbf{F}^{i}(k)^\top + \mathbf{G}^i(k)\mathbf{Q}^i(k)\mathbf{G}^{i}(k)^\top, \quad (14b)$$

$$\mathbf{P}_{ij}^-(k+1) = \mathbf{F}^i(k)\mathbf{P}_{ij}^+(k)\mathbf{F}^j(k)^\top, \quad j \in \mathcal{V} \setminus \{i\}. \quad (14c)$$

2: Update: While there are no relative measurements no update happens, i.e.,

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

When there is a relative measurement at time-step $k+1$, for example robot a makes a relative measurement of robot b , the update proceeds as below. The residual of the relative measurement and its covariance are, respectively,

$$\mathbf{r}^a = \mathbf{z}_{ab} - \mathbf{h}_{ab}(\hat{\mathbf{x}}^{a-}(k+1), \hat{\mathbf{x}}^{b-}(k+1)),$$

and

$$\begin{aligned} \mathbf{S}_{ab} &= \mathbf{R}^a(k+1) + \tilde{\mathbf{H}}_a(k+1)\mathbf{P}^{a-}(k+1)\tilde{\mathbf{H}}_a(k+1)^\top + \tilde{\mathbf{H}}_b(k+1)\mathbf{P}^{b-}(k+1)\tilde{\mathbf{H}}_b(k+1)^\top \\ &\quad - \tilde{\mathbf{H}}_b(k+1)\mathbf{P}_{ba}^-(k+1)\tilde{\mathbf{H}}_a(k+1)^\top - \tilde{\mathbf{H}}_a(k+1)\mathbf{P}_{ab}^-(k+1)\tilde{\mathbf{H}}_b(k+1)^\top. \end{aligned} \quad (15)$$

The estimation updates for the centralized EKF are:

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i-}(k+1) + \mathbf{K}_i(k+1)\mathbf{r}^a(k+1), \quad (16a)$$

$$\mathbf{P}^{i+}(k+1) = \mathbf{P}^{i-}(k+1) - \mathbf{K}_i\mathbf{S}_{ab}(k+1)\mathbf{K}_i(k+1)^\top, \quad (16b)$$

$$\mathbf{P}_{ij}^+(k+1) = \mathbf{P}_{ij}^-(k+1) - \mathbf{K}_i(k+1)\mathbf{S}_{ab}(k+1)\mathbf{K}_j(k+1)^\top, \quad (16c)$$

where $i \in \mathcal{V}$, $j \in \mathcal{V} \setminus \{i\}$ and

$$\mathbf{K}_i = (\mathbf{P}_{ib}^-(k+1)\tilde{\mathbf{H}}_b^\top - \mathbf{P}_{ia}^-(k+1)\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}^{-1}. \quad (17)$$

3: $k \leftarrow k+1$

Algorithm 2 *Interim Master D-CL*

Require: Initialization ($k = 0$): Every agent $i \in \mathcal{V}$ initializes its filter at

$$\hat{\mathbf{x}}^{i+}(0) \in \mathbb{R}^{n^i}, \mathbf{P}^{i+}(0) \in \mathbb{M}_{n^i}, \mathbf{\Phi}^i(0) = \mathbf{I}_{n^i}, \bar{\mathbf{P}}_{jl}^i(0) = \mathbf{0}_{n^l \times n^j}, j \in \mathcal{V} \setminus \{N\}, l \in \{j+1, \dots, N\}. \quad (18)$$

Iteration k

1: Propagation: Every agent $i \in \mathcal{V}$ propagates the variables below

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

2: Update: while there are no relative measurements in the network, every agent $i \in \mathcal{V}$ updates its variables as:

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i-}(k+1), \mathbf{P}^{i+}(k+1) = \mathbf{P}^{i-}(k+1), \bar{\mathbf{P}}_{jl}^i(k+1) = \bar{\mathbf{P}}_{lj}^i(k), j \in \mathcal{V} \setminus \{N\}, l \in \{j+1, \dots, N\}.$$

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

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

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

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

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 (5).

The interim master passes the following data, either directly or indirectly (by message passing), to the rest of the agents in the network:

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

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

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

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

$$\hat{\mathbf{x}}^{j+}(k+1) = \hat{\mathbf{x}}^{j-}(k+1) + \mathbf{\Phi}^j(k+1) \bar{\mathbf{D}}_i \bar{\mathbf{r}}^a, \quad (19a)$$

$$\mathbf{P}^{j+}(k+1) = \mathbf{P}^{j-}(k+1) - \mathbf{\Phi}^j(k+1) \bar{\mathbf{D}}_i \bar{\mathbf{D}}_i^\top \mathbf{\Phi}^j(k+1)^\top, \quad (19b)$$

$$\bar{\mathbf{P}}_{jl}^j(k+1) = \bar{\mathbf{P}}_{jl}^j(k) - \bar{\mathbf{D}}_j \bar{\mathbf{D}}_l^\top. \quad (19c)$$

3: $k \leftarrow k+1$

TABLE I: Complexity analysis per agent 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 .

Algorithm	Computation		Storage		Broadcast*		Message Size		Connectivity	
	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
Update per N_z relative measur.	$O(N_z \times N^2)$	$O(N_z \times N^2)$	$O(N^2)$	$O(N^2)$	$O(N_z)$	$O(N_z)$	$O(1)$	$O(1)$	interim mas- ter can reach	connected digraph
Overall worst case	$O(N^4)$	$O(N^4)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(1)$	$O(1)$	all the agents	

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

with the worst cost of $O(N)$.

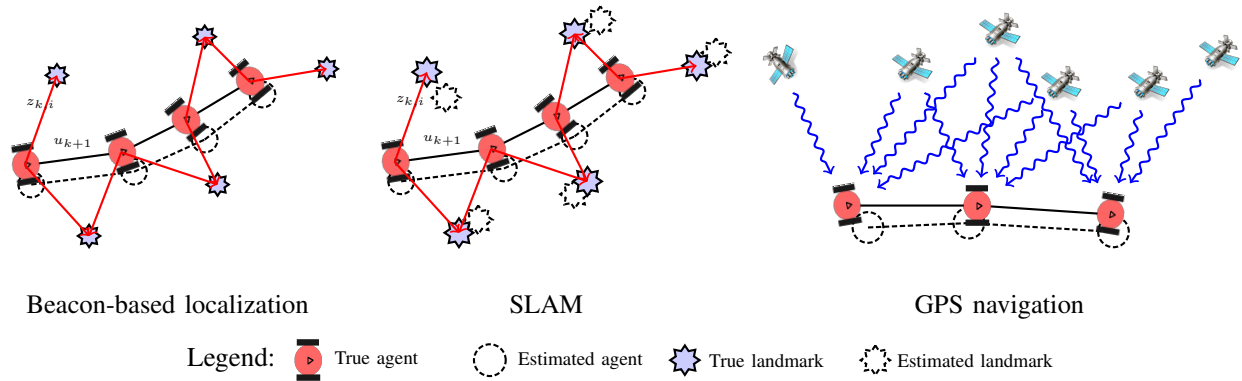


Figure 1: Schematic representation of common localization techniques for mobile platforms: In beacon-based localization, the map of the area is known and there are pre-installed beacons or landmarks with known locations and identities. By taking relative measurements with respect to these landmarks, the mobile agents can improve their localization accuracy. For operations where a priori knowledge about the environment is not available, but nevertheless, the environment contains fixed and distinguishable features that agents can measure, SLAM is normally used to localize the mobile agents. SLAM is a process by which a mobile agent can build a map of an environment and at the same time use this map to deduce its location. On the other hand, GPS navigation provides location and time information in all weather conditions, anywhere on or near the earth but it requires an unobstructed line of sight to at least four GPS satellites.

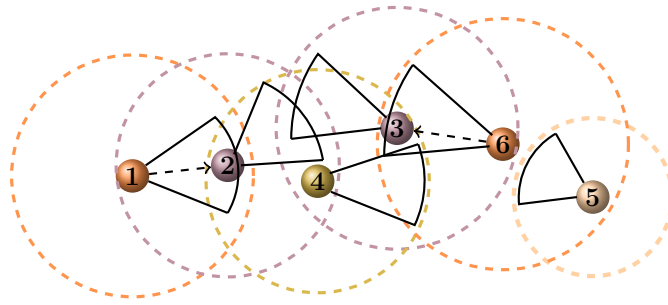


Figure 2: The spheres represent the robots and the dashed circles around them represents the communication range of the robots. The circular sectors depict the exteroceptive sensing zone of the corresponding robot. Here, robots 1 and 6 make relative measurements, respectively, of robots 2 and 3. For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

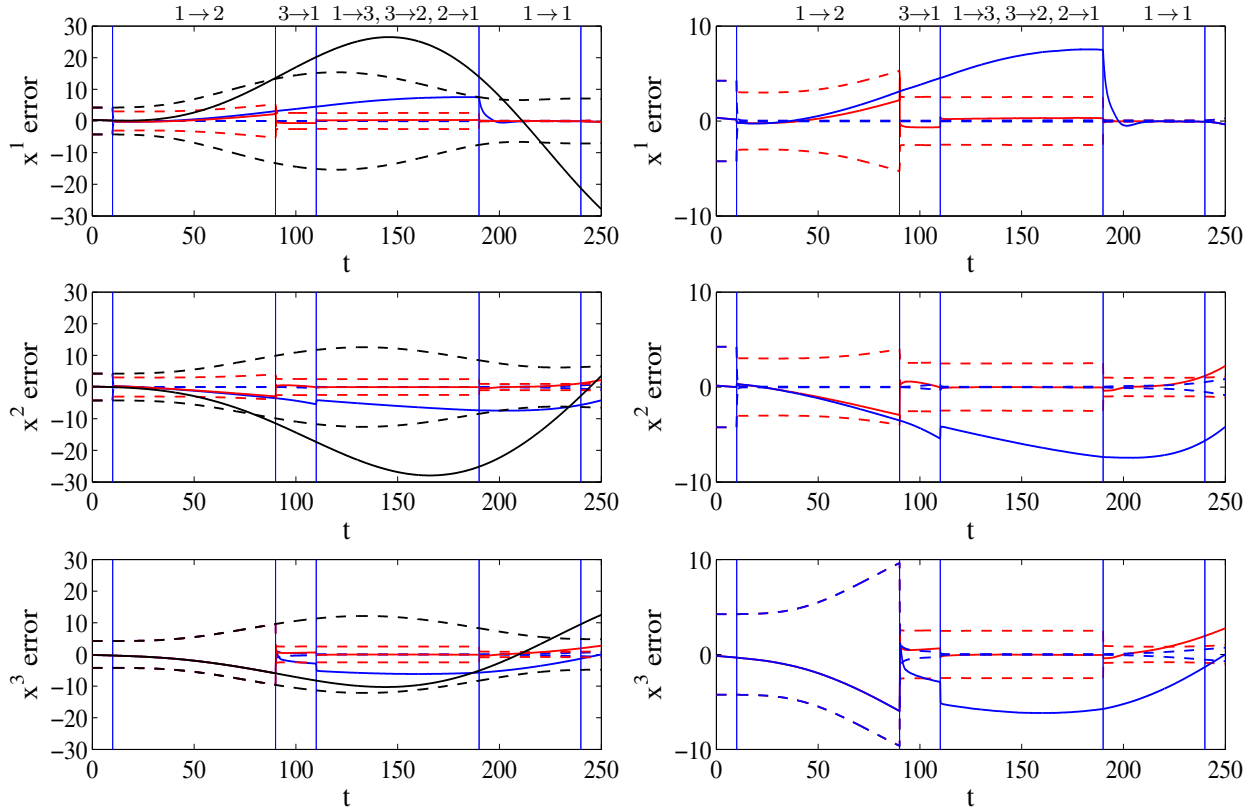


Figure 3: Estimation error (solid line) and 3σ error bound (dashed lines) in the x -coordinate variable of 3 robots moving on a flat terrain when they (a) only propagate their equations of motion using self-motion measurements (black plots), (b) employ cooperative localization ignoring past correlations between the estimations of the robots (blue plots), (c) employ cooperative localization with accurate account of past correlations (red plots). The figures in the right column are the repetition for figures in the left where the localization case (a) is removed for clearer demonstration of cases (b) and (c). Here, $a \rightarrow b$ over the time interval marked by two vertical blue lines indicates that robot a has taken a relative measurement with respect to robot b at that time interval. The symbol $a \rightarrow a$ means that robot a obtains an absolute measurement.

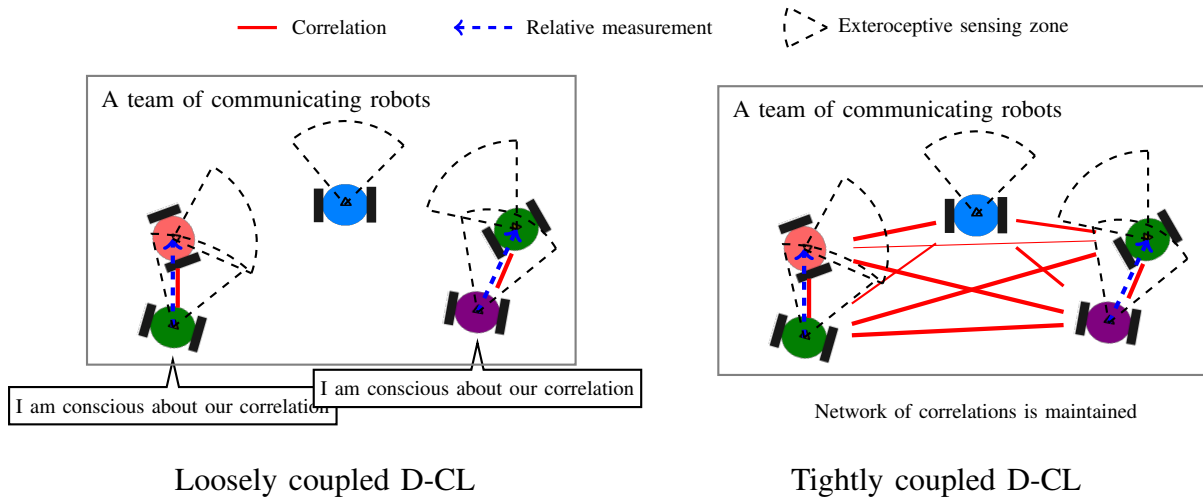


Figure 4: Schematic representation of the D-CL classification based on how the past correlations are accounted for.

Sidebar 1

Notation used throughout this document

We denote by \mathbb{M}_n , $\mathbf{0}_{n \times m}$ (when $m = 1$, we use $\mathbf{0}_n$) and \mathbf{I}_n , respectively, the set of real positive definite matrices of dimension $n \times n$, the zero matrix of dimension $n \times m$, and the identity matrix of dimension $n \times n$. We represent the transpose of matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ by \mathbf{A}^\top . The block diagonal matrix of set of matrices $\mathbf{A}_1, \dots, \mathbf{A}_N$ is $\text{Diag}(\mathbf{A}_1, \dots, \mathbf{A}_N)$. For finite sets V_1 and V_2 , $V_1 \setminus V_2$ is the set of elements in V_1 , but not in V_2 . For a finite set V we represent its cardinality by $|V|$. In the network of N robots, the local variables associated with robot i are distinguished 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. 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. The cross-covariance of the pose vectors of robots i and j is \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 as well as matrix dimensions whenever they are clear from the context. In a network of N robots, $\mathbf{p} = (\mathbf{p}^1, \dots, \mathbf{p}^N) \in \mathbb{R}^d$, $d = \sum_{i=1}^N n^i$ is the aggregated vector of local vectors $\mathbf{p}^i \in \mathbb{R}^{n^i}$.

Sidebar 2

Further Reading on EKF Cooperative Localization

CL using EKF, or EKF CL in what follows, is a convenient strategy both for analysis and implementation purposes. For a group of N homogeneous robots with the same level of uncertainty in their proprioceptive measurements that move on a flat terrain and use exteroceptive relative pose measurements, [S1] provides an analytical expression for the upper bound on the positioning uncertainty increase rate as a function of N , the odometric and orientation uncertainty for the robots, and the accuracy of a robot tracker measuring relative positions between pairs of robots. The accuracy of position estimation for groups of possibly heterogeneous robots moving on a flat terrain and performing cooperative localization is studied in [S2] where an analytical expression for the upper bound on the expected positioning uncertainty of robots is provided. This bound is determined as a function of the sensors' noise covariance and the eigenvalues of the relative position measurement graph, i.e., the weighted directed graph which represents the network of robot-to-robot exteroceptive measurements. The consistency of EKF CL from the perspective of observability is studied in [S3]. Huang et al. in [S3] analytically show that the error-state system model employed in the standard EKF CL always has an observable subspace of higher dimension than that of the actual nonlinear CL system. This results in unjustified reduction of the EKF covariance estimates in directions of the state space where no information is available, and thus leads to inconsistency. To address this problem, Huang et al. in [S3] adopt an observability-based methodology for designing consistent estimators in which the linearization points are selected to ensure a linearized system model with an observable subspace of the correct

dimension.

References

- [S1] S. I. Roumeliotis and A. I. Mourikis. Propagation of uncertainty in cooperative multirobot localization: Analysis and experimental results. *Autonomous Robots*, 17(1):1573–7527, 2004.
- [S2] A. I. Mourikis and S. I. Roumeliotis. Performance analysis of multirobot cooperative localization. *IEEE Transactions on Robotics*, 22(4):666–681, 2006.
- [S3] G. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis. Observability-based consistentekf estimators for multi-robot cooperative localization. *Autonomous Robots*, 30(1):37–58, 2011.

Sidebar 3

Covariance Intersection

Covariance intersection, proposed by Julier and Uhlmann in [S4], [S5], is a general approach to the problem of fusing correlated estimates of two or more different estimates of a random variable, each represented by its own estimated mean and covariance. Through a fusion process which resembles the update step of the Kalman filter, the covariance intersection algorithm, despite the lack of explicit knowledge about the correlation between the estimates, produces a posterior fused estimation with a covariance that guarantees consistency under the assumption of Gaussian noise. Examples of estimation divergence as a result of a naive approach of fusing the state estimations without taking into account the intrinsic correlation between them are given in [S4], [14].

References

- [S4] S. J. Julier and J. K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *American Control Conference*, pages 2369–2373, Albuquerque, NM, 1997.
- [S5] S. J. Julier and J. K. Uhlmann. Simultaneous localisation and map building using split covariance intersection. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 1257–1262, Maui, HI, 2001.