Cooperative localization under message dropouts via a partially decentralized EKF scheme

Solmaz S. Kia Stephen Rounds Sonia Martínez

Abstract— **For a team of mobile robots with limited onboard resources, we propose a partially decentralized implementation of an extended Kalman filter for cooperative localization. In the proposed algorithm, unlike a fully centralized scheme that requires, at each timestep, information from the entire team to be gathered together and be processed by a single device, we only require that the robots communicate with a central command unit at the time of a measurement update. In addition, the computational and storage cost per robot in terms of the size of the team is reduced to $O(1)$. Moreover, the algorithm is robust to occasional in-network communication link failures while the estimation update of the robots receiving the update message is of minimum variance. We demonstrate the performance of the algorithm in simulations.**

Keywords: Cooperative localization; limited onboard resources; message dropouts.

I. Introduction

The objective of cooperative localization (CL) is to increase the localization accuracy of a team of mobile robots by *jointly* estimating their locations using intra-team relative measurements. This technique, unlike classical beacon-based localization algorithms [1] or fixed feature-based Simultaneous Localization and Mapping algorithms [2], does not rely on external features of the environment. As such, this approach is an appropriate localization strategy in applications that take place in a priori inaccessible and uncharted environments where features are dynamic or not revisited as well as those applications with no or intermittent GPS access. A major concern in developing any CL algorithm with an efficient communication strategy is how to keep an accurate account of the intrinsic cross-correlations of state estimations without resorting to all-to-all multi-robot communications at each time-step. Accounting for the cross-correlations is crucial for both filter consistency and also expanding the benefit of an update of a robot-to-robot measurement to the entire team. The problem becomes more challenging if in-network communications fail due to external events such as obstacle blocking or limited communication ranges. In this paper, we look at such issue by proposing a partially decentralized filtering strategy.

Fully centralized CL schemes, *at each time-step*, gather and process information from the entire team at a single device, either a leader robot or a fusion center (FC), and broadcast back the estimated location results to each robot [3], [4]. Various decentralized CL (D-CL) algorithms have also been proposed in the literature. In [5], a suboptimal algorithm where only the robot obtaining the relative measurement updates its states is proposed. Here, a bank of Extended Kalman Filters (EKFs) together with an accurate book-keeping of robots involved in previous updates is maintained by each robot to produce consistent estimates. Although this method does not impose a particular in-network communication graph, its computational complexity, large memory demand, and the growing size of information needed at each update time are the main drawbacks. Alternatively, the computation of components of a centralized CL can be distributed among team members. For example, this decentralization can be conducted as a multi-centralized CL, wherein each robot broadcasts its own information to the entire team, which later reproduce the centralized pose estimates acting as a 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 CL algorithm is proposed in [7] where propagation stage is fully decentralized by splitting each cross-covariance term between the corresponding two robots. However, at update times, the separated parts should be combined, requiring an all-to-all robot communication. Another D-CL algorithm based on decoupling the propagation stage of an EKF CL using new intermediate variables is proposed in [8]. But here, unlike [7], at update stage, each robot can locally reproduce the updated pose estimate and covariance of the centralized EKF after receiving an update message only from the robot that has made the relative measurement. Subsequently, [9] presents a maximum-a-posteriori (MAP) D-CL algorithm in which all the robots in the team calculate parts of the centralized CL.

The algorithms above all assume that communication messages are delivered, as prescribed, perfectly all the time. A D-CL approach equivalent to a centralized CL, when possible, that handles both limited communication ranges and time-varying communication graphs is proposed in [10]. This technique uses an information transfer scheme wherein each robot broadcasts all its locally available information (the past and present measurements, as well as past measurements previously received from other robots) to every robot within its communication radius at each time-step. The main drawback of this algorithm is its high communication and storage cost. CL techniques for non-Gaussian noises are discussed in [11], [12] but they do not address the communication failure.

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

Motivated by the limited on-board resources in micro-robots, and at the same time, with a desire to eliminate the communication per time-step requirement of fully centralized CLs, we propose a partially decentralized CL strategy with fully decoupled propagation stage and centralized update through a central command unit (CCU). Our algorithm is an implementation of an EKF for CL and builds on the EKF decoupling strategy proposed in [8]. The fully decentralized algorithm of [8] requires an $O(N^2)$ storage and $O(N^2)$ per measurement update processing cost per robot, where N is the size of the cooperative robotic team. These costs can be reduced to $O(N)$ with the penalty of bigger communication message sizes. Without such a cost, maintaining the intrinsic cross-covariances of the CL strategy in a fully decentralized manner is not possible. In the proposed algorithm here, we put the CCU in charge of maintaining the cross-covariances and the calculation of the update gains, and reduce the storage and processing cost per robot to $O(1)$. Also by fully decoupling the propagation stage, we reduce the communication incidences to exteroceptive measurement update times. Our next contribution is to show that the proposed partially D-CL strategy is also robust to occasional message dropouts in the network, which is not the case in the previous fully decentralized scheme of [8].

Notations: the set of $n\times n$ real positive definite matrices is $\mathbb{M}_n$. The $n\times m$ zero matrix (when $m=1$, we use $\mathbf{0}_n$) is $\mathbf{0}_{n\times m}$ while the $n\times n$ identity matrix is $\mathbf{I}_n$. The transpose of matrix $\mathbf{A}\in\mathbb{R}^{n\times m}$ is $\mathbf{A}^\top$. The block diagonal matrix of set of matrices $\mathbf{A}_1,\dots,\mathbf{A}_N$ is $\operatorname{Diag}(\mathbf{A}_1,\cdots,\mathbf{A}_N)$. For finite sets V_1 and V_2, $V_1\backslash V_2$ is the set of elements in V_1, but not in V_2. The cardinality of a finite set V is $|V|$. In a team of N robots, the local variables of 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. We use the term *cross-covariance* to refer to the correlation terms between two robots in the covariance matrix of the entire team, and demonstrate the cross-covariance of the pose vectors of robots i and j by $\mathbf{P}_{ij}$. We denote the propagated and updated variables, say $\hat{\mathbf{x}}^i$, at time-step k by $\hat{\mathbf{x}}^{i\text{-}}(k)$ 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. The aggregated vector of local vectors $\mathbf{p}^i\in\mathbb{R}^{n^i}$ is $\mathbf{p}=(\mathbf{p}^1,\dots,\mathbf{p}^N)\in\mathbb{R}^d$, $d=\sum_{i=1}^N n^i$.

II. Description of the mobile robot team

We consider a team of N mobile robots with communication, processing and measurement capabilities. The robots are only communicating with a CCU that oversees the operation. This CCU can also be a team member with greater processing and storage capabilities. The assumption is that the CCU can reach every robot in the team, but the communication lines can be interrupted from time to time. Every robot has a detectable unique identity (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 proprioceptive sensors every robot $i\in\mathcal{V}$ 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 robot 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 robot $i\in\mathcal{V}$. The robotic team can be heterogeneous, nevertheless, the collective motion equation of the team reads

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

where, $\mathbf{f}(\mathbf{x},\mathbf{u})=(\mathbf{f}^1(\mathbf{x}^1,\mathbf{u}^1),\cdots,\mathbf{f}^N(\mathbf{x}^N,\mathbf{u}^N))$ and $\mathbf{g}(\mathbf{x})=\operatorname{Diag}(\mathbf{g}^1(\mathbf{x}^1),\cdots,\mathbf{g}^N(\mathbf{x}^N))$. The process noises $\mathbf{n}^i$, $i\in\mathcal{V}$, are independent zero-mean white Gaussian processes with a known positive definite variance $\mathbf{Q}^i(k)=E[\mathbf{n}^i(k)^\top\mathbf{n}^i(k)]$. Every robot also carries exteroceptive sensors to monitor the environment to detect, uniquely, the other robots in the team and take relative measurements from them, e.g., range or bearing or both. We model the relative measurement collected by robot i from robot j as

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

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 $\mathbf{R}^i(k)=E[\boldsymbol{\nu}^i(k)^\top\boldsymbol{\nu}^i(k)]$. All 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 CL algorithm

In this section, we review the centralized EKF CL algorithm of [7], which is a straightforward application of EKF over the collective motion model (1) using the relative measurement model (2). The propagation stage of this algorithm is

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

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

where $\mathbf{F}=\operatorname{Diag}(\mathbf{F}^1,\cdots,\mathbf{F}^N)$, $\mathbf{G}=\operatorname{Diag}(\mathbf{G}^1,\cdots,\mathbf{G}^N)$ and $\mathbf{Q}=\operatorname{Diag}(\mathbf{Q}^1,\cdots,\mathbf{Q}^N)$, with $\mathbf{F}^i=\frac{\partial}{\partial\mathbf{x}^i}\mathbf{f}(\hat{\mathbf{x}}^{i+}(k),\mathbf{u}^i(k))$ and $\mathbf{G}^i=\frac{\partial}{\partial\mathbf{x}^i}\mathbf{g}(\hat{\mathbf{x}}^{i+}(k))$, for all $i\in\mathcal{V}$.

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

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

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

where

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

Then, the Kalman gain, collective pose update and covariance update equations for the team, respectively, are:

$$\mathbf{K}(k+1)=\mathbf{P}^{\text{-}}(k+1)\mathbf{H}_{ab}(k+1)^\top{\mathbf{S}_{ab}}^{-1}. \tag{6a}$$

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

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

Algorithm 1 Centralized EKF CL

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

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

Iteration k

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

$$\hat{\mathbf{x}}^{i-}(k+1)=\mathbf{f}^i(\hat{\mathbf{x}}^{i+}(k),\mathbf{u}^i(k)),\tag{7a}$$
$$\mathbf{P}^{i-}(k+1)=\mathbf{F}^i(k)\mathbf{P}^{i+}(k)\mathbf{F}^i(k)^\top+\mathbf{G}^i(k)\mathbf{Q}^i(k)\mathbf{G}^i(k)^\top,\tag{7b}$$
$$\mathbf{P}^{-}_{ij}(k+1)=\mathbf{F}^i(k)\mathbf{P}^{+}_{ij}(k)\mathbf{F}^j(k)^\top,\quad j\in\mathcal{V}\backslash\{i\}.\tag{7c}$$

2: Update: While there are no relative measurements no update happens:

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

If a robot a takes a relative measurement from robot b, proceed by steps below. The measurement residual and its covariance are, respectively, (4a) and

$$\begin{aligned}\mathbf{S}_{ab}=\;&\mathbf{R}^a(k+1)+\tilde{\mathbf{H}}_a(k+1)\mathbf{P}^{a-}(k+1)\tilde{\mathbf{H}}_a(k+1)^\top\\&+\tilde{\mathbf{H}}_b(k+1)\mathbf{P}^{b-}(k+1)\tilde{\mathbf{H}}_b(k+1)^\top\\&-\tilde{\mathbf{H}}_b(k+1)\mathbf{P}^{-}_{ba}(k+1)\tilde{\mathbf{H}}_a(k+1)^\top\\&-\tilde{\mathbf{H}}_a(k+1)\mathbf{P}^{-}_{ab}(k+1)\tilde{\mathbf{H}}_b(k+1)^\top.\end{aligned}\tag{8}$$

The estimation updates for the centralized EKF are:

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

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

$$\mathbf{K}_i=(\mathbf{P}^{-}_{ib}(k+1)\tilde{\mathbf{H}}_b^\top-\mathbf{P}^{-}_{ia}(k+1)\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-1}.\tag{10}$$

3: $k\leftarrow k+1$

Let $\mathbf{K}=\left[\mathbf{K}_1^\top,\cdots,\mathbf{K}_N^\top\right]^\top$, 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, we can express the collective centralized EKF CL in terms of its robot-wise components, as shown in Algorithm 1. To process multiple synchronized measurements, we use *sequential updating* (c.f. e.g., [13, ch. 3],[14]). Note that, because of the inherent coupling in cross-covariance terms (7c) and (9c), Algorithm 1 can not be implemented in a decentralized manner. Next, we propose an implement of the EKF CL algorithm where the propagation stage is implemented locally and the updates are dictated by a CCU.

IV. Partially decentralized implementation of the EKF for CL

In this section, we present an implementation of the EKF for CL where the propagation stage is fully decentralized but the updates are carried out in centralized manner. Our scheme builds on the decoupling method of [8] which results in a decentralized implementation of Algorithm 1 where the robot making the relative measurement is designated as the *interim master* and provides the rest of the team with the information they need to update their pose, $\hat{\mathbf{x}}^{i+}$, and covariance, $\mathbf{P}^{i+}$ matchin (9a) and (9b). The algorithm in [8] results in an $O(N^2)$ storage and $O(N^2\times N_z)$, processing cost per robot with N_z the total number of relative measurement in the team in a given time. The following partially decentralized implementation reduces this cost to $O(1)$ per robot by using a CCU to maintain team cross-covariances, which is the source of high processing and storage costs.

We start by reviewing the EKF decoupling approach of [8], which uses the following assumption that is valid for many mobile robot models.

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

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:

$$\mathbf{\Phi}^i(k+1)=\mathbf{F}^i(k)\mathbf{\Phi}^i(k).\tag{11}$$

Then, we write the propagated cross-covariances (7c) as:

$$\mathbf{P}^{-}_{ij}(k+1)=\mathbf{\Phi}^i(k+1)\bar{\mathbf{P}}_{ij}(k)\mathbf{\Phi}^j(k+1)^\top,\tag{12}$$

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 $k+1$, (12) results in $\bar{\mathbf{P}}_{ij}(k+1)=\bar{\mathbf{P}}_{ij}(k)$. Next, we derive an expression for $\bar{\mathbf{P}}_{ij}(k+1)$ when there is a in-network relative measurement at time $k+1$, such that at time $k+2$ we could write $\mathbf{P}^{-}_{ij}(k+2)=\mathbf{\Phi}^i(k+2)\bar{\mathbf{P}}_{ij}(k+1)\mathbf{\Phi}^j(k+2)^\top$. For this notice that we can rewrite the update equations (8) and (10) of the centralized CL algorithm by replacing the cross-covariance terms by (12):

$$\begin{aligned}\mathbf{S}_{ab}=\;&\mathbf{R}^a+\tilde{\mathbf{H}}_a\mathbf{P}^{a-}(k+1)\tilde{\mathbf{H}}_a^\top+\tilde{\mathbf{H}}_b\mathbf{P}^{b-}(k+1)\tilde{\mathbf{H}}_b^\top-\\&\tilde{\mathbf{H}}_a\mathbf{\Phi}^a(k+1)\bar{\mathbf{P}}_{ab}(k)\mathbf{\Phi}^b(k+1)^\top\tilde{\mathbf{H}}_b^\top-\\&\tilde{\mathbf{H}}_b\mathbf{\Phi}^b(k+1)\bar{\mathbf{P}}_{ba}(k)\mathbf{\Phi}^a(k+1)^\top\tilde{\mathbf{H}}_a^\top,\end{aligned}\tag{13}$$

and the Kalman gain is

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

where

$$\bar{\mathbf{D}}_i=(\bar{\mathbf{P}}_{ib}(k){\mathbf{\Phi}^b}^\top\tilde{\mathbf{H}}_b^\top-\bar{\mathbf{P}}_{ia}(k){\mathbf{\Phi}^a}^\top\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-\frac{1}{2}},\ i\in\mathcal{V}\backslash\{a,b\},\tag{14a}$$
$$\bar{\mathbf{D}}_a=(\bar{\mathbf{P}}_{ab}(k){\mathbf{\Phi}^b}^\top\tilde{\mathbf{H}}_b^\top-(\mathbf{\Phi}^a)^{-1}\mathbf{P}^{a-}\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-\frac{1}{2}},\tag{14b}$$
$$\bar{\mathbf{D}}_b=((\mathbf{\Phi}^b)^{-1}\mathbf{P}^{b-}\tilde{\mathbf{H}}_b^\top-\bar{\mathbf{P}}_{ba}(k){\mathbf{\Phi}^a}^\top\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-\frac{1}{2}}.\tag{14c}$$

Notice that due to Assumption 1, $\mathbf{\Phi}^i(k)$, 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

$$\bar{\mathbf{P}}_{ij}(k+1)=\bar{\mathbf{P}}_{ij}(k)-\bar{\mathbf{D}}_i\bar{\mathbf{D}}_j^\top.\tag{15}$$

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 (12) 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 $\mathbf{\Phi}^i(k)$'s and $\bar{\mathbf{P}}_{ij}$'s. Let $\bar{\mathbf{r}}^a=(\mathbf{S}_{ab})^{-\frac{1}{2}}\mathbf{r}^a$, then the updated state estimate and covariance matrix in the new variables reads as, for $i\in\mathcal{V}$,

$$\begin{aligned}\hat{\mathbf{x}}^{i+}(k+1)&=\hat{\mathbf{x}}^{i-}(k+1)+\mathbf{\Phi}^i(k+1)\,\bar{\mathbf{D}}_i\,\bar{\mathbf{r}}^a,\\\mathbf{P}^{i+}(k+1)&=\mathbf{P}^{i-}(k+1)-\mathbf{\Phi}^i(k+1)\bar{\mathbf{D}}_i\bar{\mathbf{D}}_i^\top\mathbf{\Phi}^i(k+1)^\top,\end{aligned}\tag{16}$$

Using the decompositions above, our proposed partially decentralized CL algorithm is as follows. Every robot $i\in\mathcal{V}$ maintains and propagates it propagated state estimation (7a) and its corresponding covariance matrix (7b), as well as, variable $\mathbf{\Phi}^i$ (11). Notice that all these variables depend only on local data. Therefore, the propagation stage is fully decoupled. The CCU is in charge of maintaining and updating $\bar{\mathbf{P}}_{ij}$'s. When there is a relative measurement in the network, say robot a takes relative measurement from robot b, robot a informs the CCU. Then, the CCU starts the update procedure by taking the following actions. It acquires ($\mathbf{z}_{ab} \in \mathbb{R}^{n^a_z}$, $\hat{\mathbf{x}}^{a\text{-}}(k+1) \in \mathbb{R}^{n^a}$, $\mathbf{\Phi}^a(k+1) \in \mathbb{R}^{n^a \times n^a}$, $\mathbf{P}^{a\text{-}}(k+1) \in \mathbb{M}_{n^a}$) from robot a and ($\hat{\mathbf{x}}^{b\text{-}}(k+1) \in \mathbb{R}^{n^b}$, $\mathbf{\Phi}^b(k+1) \in \mathbb{R}^{n^b \times n^b}$, $\mathbf{P}^{b\text{-}}(k+1) \in \mathbb{M}_{n^b}$) from robot b. Then, using this information, which we refer to it as *landmark-message*, along with its locally maintained $\bar{\mathbf{P}}_{ij}$'s, it calculates $\mathbf{r}_a$, $\mathbf{S}_{ab}$ and $\bar{\mathbf{D}}_i$, $i \in \mathcal{V}$, from respectively, (4a), (13) and (14). Then, the CCU sends to each robot $i \in \mathcal{V}$ its corresponding update message $(\bar{\mathbf{D}}_i\,\bar{\mathbf{r}}^a, \bar{\mathbf{D}}_i\bar{\mathbf{D}}_i^\top)$ so that the robot can update its local estimates using (16). It also updates its local $\bar{\mathbf{P}}_{ij}$'s using (15), for all $i \in \mathcal{V}\backslash\{N\}$ and $j \in \{i+1,\cdots,N\}$–because of the symmetry of the covariance matrix of the network we only need to save, e.g., the upper triangular part of this matrix. Algorithm 2 presents this partially decentralized implementation of EKF for CL when there is only one relative measurement incident at a time. This algorithm operates based on the assumption that at the time of measurement update, all the robots can receive the update message of the CCU. This requirement is relaxed in proceeding section, where we study the robustness of our proposed algorithm to message dropouts.

To include absolute measurements in Algorithm 2 the CCU only needs the information of the robot that has obtained the absolute measurement. It proceeds with the similar updating procedure as outlined above and issues the corresponding update message $(\bar{\mathbf{D}}_i\,\bar{\mathbf{r}}^a, \bar{\mathbf{D}}_i\bar{\mathbf{D}}_i^\top)$ to every robot $i \in \mathcal{V}$.

For multiple synchronized measurements, we use the sequential updating procedure. One can expect that the updating order must not dramatically change the results (cf. [14, page 104] and references therein) Here, we assume

Assumption 2: CCU has a pre-specified *sequential-updating-order* guideline, which indicates the priority order for implementing the measurement update.

Remark 4.1 (Multiple synchronized relative measurements): For clarity of exposition, first, following [14], we briefly review the sequential updating in Kalman filters. Let N_s synchronous sensor measurements at time k be represented by $\mathbf{z}_j$, $j \in \{1,\cdots,N_s\}$. Let

$$\hat{\mathbf{x}}^+(k,0) = \hat{\mathbf{x}}^-(k), \quad \mathbf{P}^+(k,0) = \mathbf{P}^-(k).$$

The update at time k is $\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^+(k,N_s)$ and $\mathbf{P}^+(k) = \mathbf{P}^+(k,N_s)$, obtained from the following procedure:

$$\begin{aligned}\hat{\mathbf{x}}^+(k,j) =& \hat{\mathbf{x}}^+(k,j-1) + \mathbf{K}(k,j)\mathbf{r}(k,j),\\ \mathbf{P}^+(k,j) =& \mathbf{P}^+(k,j-1) - \mathbf{K}(k,j)\mathbf{S}(k,j)\mathbf{K}(k,j)^\top,\end{aligned}$$

for $j \in \{1,\cdots,N_s\}$, where $\mathbf{r}(k,j)$, $\mathbf{S}(k,j)$, and $\mathbf{K}(k,j)$ are, respectively, the measurement innovation, the innovation covariance and the Kalman gain calculated using $\hat{\mathbf{x}}^+(k,j-1)$ and $\mathbf{P}^+(k,j-1)$. Implementing the decomposition introduced above, the sequential updating by CCU is described in Algorithm 3, where we used the following notation. Let $\mathcal{V}_{\text{M}}$ be the set of the robots that have made an exteroceptive measurement at time $k+1$, $\mathcal{V}_{\text{L}}(i)$ be the landmark robots of robot $i \in \mathcal{V}_{\text{M}}$, and $\bar{\mathcal{V}}$ be the set of all the robots taken relative measurements and landmark robots. We assume that the robots making measurements inform the CCU and indicate to CCU what their landmark robots are. Therefore, the CCU knows $\mathcal{V}_{\text{M}}$ and $\mathcal{V}_{\text{L}}(i)$'s, and sorts both of these sets according to the it's sequential-updating-order guideline. □

Algorithm 2 *Partially D-CL*

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

$$\hat{\mathbf{x}}^{i+}(0) \in \mathbb{R}^{n^i}, \quad \mathbf{P}^{i+}(0) \in \mathbb{M}_{n^i}, \quad \mathbf{\Phi}^i(0) = \mathbf{I}_{n^i}.$$

The CCU initializes

$$\bar{\mathbf{P}}^i_{ij}(0) = \mathbf{0}_{n^i\times n^j},\; i \in \mathcal{V}\backslash\{N\}, \quad j \in \{i+1,\cdots,N\}.$$

Iteration k

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

$$\begin{aligned}&\hat{\mathbf{x}}^{i\text{-}}(k+1) = \mathbf{f}^i(\hat{\mathbf{x}}^{i+}(k), \mathbf{u}^i(k)),\\ &\mathbf{\Phi}^i(k+1) = \mathbf{F}^i(k)\mathbf{\Phi}^i(k),\\ &\mathbf{P}^{i\text{-}}(k+1) = \mathbf{F}^i(k)\mathbf{P}^{i+}(k)\mathbf{F}^i(k)^\top + \mathbf{G}^i(k)\mathbf{Q}^i(k)\mathbf{G}^i(k)^\top.\end{aligned} \tag{17}$$

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

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i\text{-}}(k+1), \quad \mathbf{P}^{i+}(k+1) = \mathbf{P}^{i\text{-}}(k+1), \tag{18}$$

and the CCU proceeds with

$$\bar{\mathbf{P}}^i_{ij}(k+1) = \bar{\mathbf{P}}^i_{ij}(k), \quad j \in \mathcal{V}\backslash\{i\}. \tag{19}$$

If there is a robot a that makes a measurement with respect to another robot b, then robot a informs the CCU. The CCU asks for the following information from robot a and b, respectively,

$$\begin{aligned}\textit{Landmark-message}^a &= \Big(\mathbf{z}_{ab}, \hat{\mathbf{x}}^{a\text{-}}(k+1), \mathbf{P}^{b\text{-}}(k+1), \mathbf{\Phi}^a(k+1)\Big),\\ \textit{Landmark-message}^b &= \Big(\hat{\mathbf{x}}^{b\text{-}}(k+1), \mathbf{P}^{b\text{-}}(k+1), \mathbf{\Phi}^b(k+1)\Big).\end{aligned} \tag{20}$$

Given the *Landmark-message*, the CCU calculates

$$\begin{aligned}\mathbf{S}_{ab} =& \mathbf{R}^a + \tilde{\mathbf{H}}_a\mathbf{P}^{a\text{-}}\tilde{\mathbf{H}}_a^\top + \tilde{\mathbf{H}}_b^\top\mathbf{P}^{b\text{-}}\tilde{\mathbf{H}}_b\\ &- \tilde{\mathbf{H}}_a\mathbf{\Phi}^a\bar{\mathbf{P}}_{ab}\mathbf{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top - \tilde{\mathbf{H}}_b\mathbf{\Phi}^b\bar{\mathbf{P}}_{ba}\mathbf{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top,\end{aligned} \tag{21}$$

as well as $\mathbf{r}^a$ and $\bar{\mathbf{D}}_i$'s using (4a), (14), respectively. It obtains $\bar{\mathbf{r}}^a = (\mathbf{S}_{ab})^{-\frac{1}{2}}\mathbf{r}^a$ and then passes the following data to every robot $i \in \mathcal{V}$ in the network:

$$\textit{update-message}^i = (\bar{\mathbf{D}}_i\,\bar{\mathbf{r}}^a, \bar{\mathbf{D}}_i\bar{\mathbf{D}}_i^\top).$$

Every robot $i \in \mathcal{V}$, upon receiving its respective *update-message*^{i}, updates its state estimate and the corresponding covariance

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i\text{-}}(k+1) + \mathbf{\Phi}^i(k+1)\textit{update-message}^i(1), \tag{22a}$$

$$\mathbf{P}^{i+}(k+1) = \mathbf{P}^{i\text{-}}(k+1) - \mathbf{\Phi}^i(k+1)\textit{update-message}^i(2)\mathbf{\Phi}^i(k+1)^\top. \tag{22b}$$

The CCU updates its local variables, for $i \in \mathcal{V}\backslash\{N\}$, $j \in \{i+1,\cdots,N\}$:

$$\bar{\mathbf{P}}_{ij}(k+1) = \bar{\mathbf{P}}^i_{ij}(k) - \bar{\mathbf{D}}_i\bar{\mathbf{D}}_j^\top. \tag{23}$$

3: $k \leftarrow k+1$

Similar to the decentralized implementation of [8], Algorithm 2 is robust to permanent team member dropouts. The CCU only suffers from a processing and communicational cost until it can confirm that the dropout is permanent.

V. Accounting for in-network message dropouts

In this section, we study the robustness of Algorithm 2 against occasional communication link failures between robots and the CCU. We show that this operation is still a minimum variance update when some of the robots miss to receive the message of the CCU. Here, we assume that the two robots involved in a relative measurement can both communicate with the CCU at the same time otherwise, we discard that measurement. We base our study on analyzing a fully centralized EKF for CL in which the FC fails to

Algorithm 3 CCU's sequential updating procedure for multiple in-network measurement at time $k+1$

Require: Initialization ($j=0$): The CCU obtains the following information from robot $a \in \mathcal{V}_{\mathrm{M}}$ and all of its landmarks $b \in \mathcal{V}_{\mathrm{L}}(a)$,

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

The CCU initializes the following variables

$$\hat{\mathbf{x}}^{+i}(k+1,0) = \hat{\mathbf{x}}^{\text{-}i}(k+1), \quad \forall i \in \bar{\mathcal{V}},$$
$$\hat{\mathbf{P}}^{+i}(k+1,0) = \mathbf{P}^{\text{-}i}(k+1), \quad \forall i \in \bar{\mathcal{V}},$$
$$\bar{\mathbf{P}}_{il}(k+1,0) = \bar{\mathbf{P}}_{il}(k), i \in \mathcal{V}\backslash\{N\},\ l \in \{i+1,\cdots,N\}.$$

Iteration j: CCU proceeds with the following calculations.

for $a \in \mathcal{V}_{\mathrm{M}}$ **do**
  for $b \in \mathcal{V}_{\mathrm{L}}(a)$ **do**
    CCU calculates $\tilde{\mathbf{H}}_a$, $\tilde{\mathbf{H}}_b$ and $\mathbf{r}^a$ using $\hat{\mathbf{x}}^{+a}(k+1,j)$ and $\hat{\mathbf{x}}^{+b}(k+1,j)$. Then, using these measurement matrices and $\hat{\mathbf{P}}^{+a}(k+1,j)$, $\hat{\mathbf{P}}^{+b}(k+1,j)$ and $\bar{\mathbf{P}}_{ab}(k+1,j)$, CCU calculates $\mathbf{S}_{ab}$ from (13) and subsequently $\bar{\mathbf{r}}(j) = (\mathbf{S}_{ab})^{-\frac{1}{2}}\mathbf{r}^a$ and $\bar{\mathbf{D}}_i(j)$ from (14) for $i \in \mathcal{V}$. Next, CCU updates the state and the covariance of all the robots in $i \in \bar{\mathcal{V}}$ as follows

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

It also updates $\bar{\mathbf{P}}_{il}$ for $i \in \mathcal{V}\backslash\{N\},\ l \in \{i+1,\cdots,N\}$ as follows

$$\bar{\mathbf{P}}_{il}(k+1,j+1) = \bar{\mathbf{P}}^j_{ij}(k+1,j) - \bar{\mathbf{D}}_i(j)\bar{\mathbf{D}}_l(j)^\top.$$

    $j \leftarrow j+1$
  end for
end for
CCU broadcasts the following update messages ($s = \sum_{i \in \mathcal{V}_{\mathrm{M}}} |\mathcal{V}_{\mathrm{L}}(i)|$)

- for robot $i \in \mathcal{V}\backslash\bar{\mathcal{V}}$

$$\textit{update-message}^i = \big(\textstyle\sum_{j=1}^{s}(\bar{\mathbf{D}}_i(j)\bar{\mathbf{r}}(j)), \sum_{j=1}^{s}(\bar{\mathbf{D}}_i(j)\bar{\mathbf{D}}_i(j)^\top)\big);$$

- for robot $i \in \bar{\mathcal{V}}$

$$\begin{aligned}\textit{update-message}^i =& ((\mathbf{\Phi}^i)^{-1}(\hat{\mathbf{x}}^{+i}(k+1,s) - \hat{\mathbf{x}}^{\text{-}i}(k+1)),\\ &-(\mathbf{\Phi}^i)^{-1}(\mathbf{P}^{+i}(k+1,s) - \mathbf{P}^{\text{-}i}(k+1))(\mathbf{\Phi}^i)^{-T}).\end{aligned}$$

---

update the estimation of some of the robots. In our partially decentralized implementation of the algorithm, these robots are those which miss the update-message of the CCU and as such they are not updating their estimates.

Let FC in a centralized CL be always able to update the estimation equation of the robots involved in a relative measurement. Without loss of generality, assume that the FC does not use the relative measurement taken by robot $a \neq N$ from robot $b \neq N$ to update the state estimation of robot N. The propagation stage of the Kalman filter is independent of the observation process and thus stays the same as the classical EKF for CL as in (7). Then, the state estimation update equations of this centralized CL algorithm are

$$\hat{\mathbf{x}}^{+}_{1:N-1}(k+1) = \hat{\mathbf{x}}^{\text{-}}_{1:N-1}(k+1) + \bar{\mathbf{K}}(k+1)\mathbf{r}^a(k+1), \tag{24}$$
$$\hat{\mathbf{x}}^{N+}(k+1) = \hat{\mathbf{x}}^{N\text{-}}(k+1), \tag{25}$$

where $\hat{\mathbf{x}}^{+}_{1:N-1} = (\hat{\mathbf{x}}^{1+}, \cdots, \hat{\mathbf{x}}^{N-1+})$. Accordingly, let the covariance matrix of the entire network be partitioned as

$$\mathbf{P} = \left[\begin{array}{c|c}\mathbf{P}_{1:N-1} & \mathbf{P}_{1:N-1,N}\\ \hline \mathbf{P}^\top_{1:N-1,N} & \mathbf{P}_N\end{array}\right].$$

Then, the CL update gain $\bar{\mathbf{K}}$ is found by $\partial\mathbf{P}_{1:N-1}/\partial\bar{\mathbf{K}} = \mathbf{0}$, which, for $i \in \mathcal{V}\backslash\{N\}$, results in

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

where $\bar{\mathbf{K}}_i$ is the component of $\bar{\mathbf{K}}$ that is used to obtain $\hat{\mathbf{x}}^{i+}$. This development is similar to the approach used to obtain *minimum variance reduced order estimators* (cf. e.g., [15, Chapter 8, Page 25]). Using this gain we obtain the following propagation equation for the covariance of robots $i \in \mathcal{V}\backslash\{N\}$ and robot N, respectively,

$$\mathbf{P}^{i+}(k+1) = \mathbf{P}^{i\text{-}}(k+1) - \bar{\mathbf{K}}_i\mathbf{S}_{ab}(k+1)\bar{\mathbf{K}}_i(k+1)^\top, \tag{26a}$$
$$\mathbf{P}^{N+}(k+1) = \mathbf{P}^{N\text{-}}(k+1). \tag{26b}$$

For cross-covariances we obtain, for $i \in \mathcal{V}$ and $j \in \mathcal{V}\backslash\{i\}$

$$\mathbf{P}^{+}_{ij}(k+1) = \mathbf{P}^{\text{-}}_{ij}(k+1) - \bar{\mathbf{K}}_i(k+1)\mathbf{S}_{ab}(k+1)\bar{\mathbf{K}}_j(k+1)^\top. \tag{27}$$

Here, we defined and used the *pseudo* gain $\bar{\mathbf{K}}_N = (\mathbf{P}^{\text{-}}_{Nb}(k+1)\tilde{\mathbf{H}}_b^\top - \mathbf{P}^{\text{-}}_{Na}(k+1)\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-1}$.

Comparing the developments above with the centralized CL where all the agents' states are updated, we observe that the state and the associated covariance update of robots $i \in \mathcal{V}\backslash\{N\}$ and also the cross-covariance update terms using the pseudo gain $\bar{\mathbf{K}}_N$ are the same. As such, the decomposition technique used to develop the partially decentralized algorithm of Section IV is valid here. Thus, we can implement exactly Algorithm 2 as is while the robots missing the update message of the CCU do not update their estimations. Therefore, this algorithm is robust to message dropouts and the estimations of the robots receiving the update message, as stated above, are minimum variance.

Interestingly, the fully decentralized algorithm of [8] is not robust to in-network message dropouts. This is due to inconsistency in the local copy of $\bar{\mathbf{P}}_{ij}$'s of the robots receiving the update message and those that do not.

VI. Comparative Performance Evaluations in Simulations

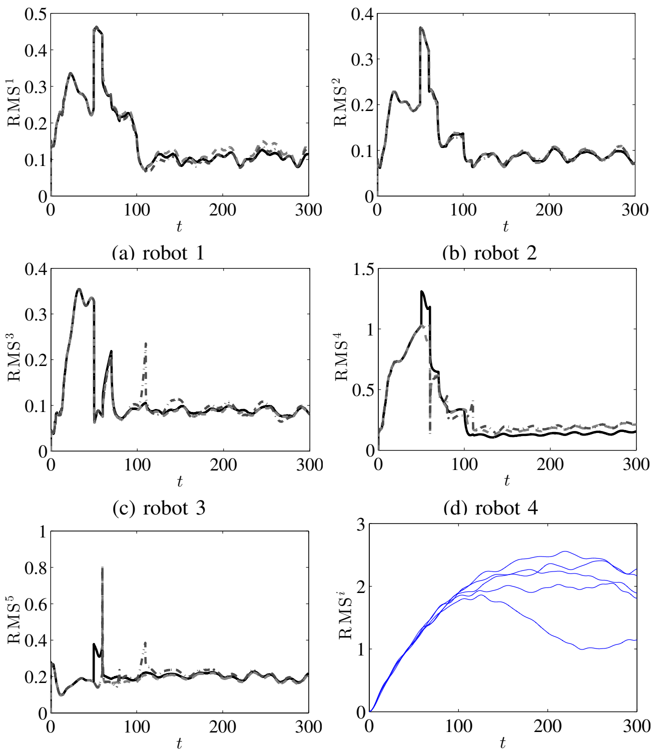
We compare the performance of the proposed partially D-CL algorithm with and without occasional communication failure in simulations. We use a team of five robots moving on a flat terrain of 25m×25m area with constant linear velocity of 0.25 m/s and the rotational velocity drawn uniformly randomly from $[0.1, 0.4]$ rad/s. The standard deviation of the linear (resp. rotational) velocity measurement noise of each robot is assume to be 5% of the linear (resp. 20% of the rotational) velocity of that robot. We assume that some robots can obtain absolute position measurement from time to time; $\mathbf{z}^i = [x^i, y^i]^\top + \boldsymbol{\nu}^i_z$ with $\sigma_{z_x} = \sigma_{z_y} = 0.1$ m. We use relative pose measurement whose contaminating noise is zero mean Gaussian with $\sigma_{z_x} = \sigma_{z_y} = 0.1$ m and $\sigma_{z_\phi} = 2$ degree, for all robots. In our test, we compare the root mean square (RMS) position and orientation error of $M = 30$ Monte Carlo simulations, with the same relative measurement scenarios. Let $\mathbf{e}^i(k) = \mathbf{x}^i - \hat{\mathbf{x}}^i(k^+)$, $i \in \{1, \cdots, 5\}$. Then, we calculate RMS using $RMS^i(k) = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\mathbf{e}^i_j(k)^\top\mathbf{e}^i_j(k)}$. Figure 1 shows the results for the measurement and communication scenarios explained in Table I.

VII. Conclusions

For a team of robots with limited computational, storage and communication resources, we proposed a partially D-CL algorithm. This localization strategy is an implementation of an EKF for CL problem where the propagation stage

TABLE I – Time table for exteroceptive measurement times and the disconnected robots. $a \to b$ indicates that robot a takes relative measurement from robot b. $a \to a$ indicates that robot a has obtained absolute measurement.

Time (sec.)	[0 50]	(50 52]	(52 60]	(60 70]	(70 72]	(72 80]	(80 100]	(100 102]	(102 110]	(110 300]
Measurements	$1 \to 2$ $2 \to 3$ $3 \to 4$ $4 \to 5$	$1 \to 2$ $\mathbf{3 \to 3}$	$1 \to 2$ $\mathbf{3 \to 3}$	$1 \to 2$ $2 \to 3$ $3 \to 4$ $4 \to 5$	$1 \to 2$ $2 \to 3$ $\mathbf{3 \to 3}$ $4 \to 5$	$1 \to 2$ $2 \to 3$ $\mathbf{3 \to 3}$ $4 \to 5$	$1 \to 2$ $2 \to 3$ $3 \to 4$ $4 \to 5$	$1 \to 2$ $\mathbf{2 \to 2}$ $3 \to 4$ $4 \to 5$	$1 \to 2$ $\mathbf{2 \to 2}$ $3 \to 4$ $4 \to 5$	$1 \to 2$ $2 \to 3$ $3 \to 4$ $4 \to 5$
Robot(s) disconnected from CCU, case 1	none	4, 5	none	none	5	none	none	4	none	none
Robot(s) disconnected from CCU, case 2	none	4, 5	4, 5	none	5	5	none	4	4	none



(a) robot 1 (b) robot 2

(c) robot 3 (d) robot 4

(e) robot 5 (f) all the robots, there is no CL

Fig. 1 – Simulation results for position RMS error for the measurement and communication scenarios described in Table I (the orientation RMS error behaves similarly and omitted for brevity). In plots (a)-(e), solid line shows the case of no communication failure; dashed (resp. dash-doted) line shows case 1 (resp. case 2) communication link failure scenario of Table I. As the simulations show the performance is very close despite occasional communication failure between robot 4 and 5 with CCU. As expected, performance deteriorates more if the link failure duration is longer. Plot (f) shows the simulation results when no CL is applied. As expected, the estimation error is much larger in this case.

is fully decentralized by decomposing the coupling terms and the updates are carried out in a CCU. In terms of the team size, this algorithm only requires $O(1)$ storage and computational cost per robot and the main computational burden of implementing the EKF for CL is carried out by the CCU. Moreover, this partially D-CL algorithm is robust to communication link failures between some robots and the CCU and the estimation update for robots that are receiving the CCU's update message is minimum variance. Here, we discarded the measurement of the robots that fail to communicate with the CCU. Our future work involves utilizing these old measurements using out-of-sequence-measurement update strategies [16] when the communication link is restored between the corresponding robot and the CCU.

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.

[3] S. I. 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. 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] 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), pp. 3761–3765, September 2014.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] C. T. 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.

[15] P. S. Maybeck, *Stochastic Models: Estimation and Control*, vol. 2. Academic Press, 1982.

[16] Y. Bar-Shalom, H. Chen, and M. Mallick, "One-step solution for the multistep out-of-sequence- measurement problem in tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 1, pp. 27–37, 2004.