UC Irvine
UC Irvine Previously Published Works

Title
Server-Assisted Distributed Cooperative Localization Over Unreliable Communication Links

Permalink
https://escholarship.org/uc/item/9g79m33j

Journal
IEEE Transactions on Robotics, 34(5)

ISSN
1552-3098

Authors
Kia, Solmaz S
Hechtbauer, Jonathan
Gogokhiya, David
et al.

Publication Date
2018

DOI
10.1109/tro.2018.2830411

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

A partially decentralized EKF scheme for cooperative localization over unreliable communication links

Solmaz S. Kia Sonia Martínez

arXiv:1608.00609v1 [cs.RO] 1 Aug 2016

Abstract

This paper reports a partially decentralized implementation of an Extended Kalman filter for the cooperative localization of a team of mobile robots with limited onboard resources. 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, our algorithm only requires that the robots communicate with a central command unit at the time of a measurement update. Every robot only needs to propagate and update its own state estimate, while the central command unit maintains track of cross-covariances. Therefore, the computational and storage cost per robot in terms of the size of the team is of order $O(1)$. Moreover, when the system model is linear the algorithm is robust to occasional in-network communication link failures while the updated estimates of the robots receiving the update message are of minimum variance at that given timestep. For problems with nonlinear robot models, our algorithm under message drop-out provides a suboptimal solutions because of the linearization approximation similar to the Extended Kalman filter model. We demonstrate the performance of the algorithm in simulation.

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.

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 the Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA 92093, USA, soniamd@ucsd.edu

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 (see [3] for further details). 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 address such an 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 [4], [5]. Various decentralized CL (DCL) algorithms have also been proposed in the literature. In [6], 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. Other 'loosely coupled' DCL strategies based on covariance intersection fusion method, where only either the robot that has takes the measurement or the landmark robot cooperatively update their state estimates, are proposed in [7], [8], [9], [10]. Although these loosely coupled cooperative localization strategies do not impose any restrictive communication topology on the team, they are conservative by nature, because they do not enable other agents in the network to fully benefit from measurement updates. Moreover, even though the covariance intersection method produces consistent estimates for a loosely coupled DCL strategy, this method is known to produce overly conservative estimates.

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 [11]. Besides a high-processing cost for each robot, this scheme requires all-to-all robot communication at the time of each information exchange. A DCL algorithm distributing computations of an EKF CL algorithm is proposed in [12] 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. Other DCL algorithm based on decoupling the propagation stage of an EKF CL using new intermediate variables is proposed in [13] and [14]. Unlike [12], in [13] 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. In [14] at update times only the two robots involved in the relative measurement need to communicate. However, in both of these algorithms, for a team of N robots, each robot incurs an $O(N^2)$ processing and storage cost as they need to evolve a variable of size of the entire covariance matrix of the robotic team. Subsequently, [15] presents a maximum-a-posteriori (MAP) DCL 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 DCL approach equivalent to a centralized CL, when possible, that handles both limited communication ranges and time-varying communication graphs is proposed in [16]. 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. In another approach towards DCL, a single-beacon cooperative acoustic localization algorithm is proposed in [17] for underwater vehicles. The reported algorithm in [17] is a decentralized extended information filter that uses ranges and state information from a single reference source (the server) with higher navigation accuracy to improve localization and navigation of underwater vehicle(s) (the client(s)) independent from one another.

Motivated by the limited onboard 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 gain calculation through a central command unit (CCU). Our algorithm is an implementation of an EKF for CL and builds on our EKF decoupling strategy previously proposed in [3]. The fully decentralized algorithm of [3] 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 our new partially decentralized algorithm, CCU is in charge of maintaining the cross-covariances and the calculation of the update gains. Therefore, the storage and processing cost per robot reduces to $O(1)$ as every robot only propagates and updates its own pose estimates. 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 DCL strategy is also robust to occasional message dropouts in the network, which is not the case in the previous fully decentralized scheme of [3]. A preliminary version of our work was presented in [18].

Notations: the set of $n\times n$ real positive definite matrices is $\mathbb{S}^n_{>0}$. The $n\times m$ zero matrix is $\mathbf{0}_{n\times m}$ (when $m=1$, we use $\mathbf{0}_n$), 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 $\mathrm{Diag}(\mathbf{A}_1,\cdots,\mathbf{A}_N)$. For block partitioned matrix $\mathbf{A}$, $\mathbf{A}_{i:j,k:l}$ where $i<j$ and $k<l$, indicates a submatrix of $\mathbf{A}$ consisted of the block in the intersection of rows i to j and the columns k to l. 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 $\mathbf{P}_{i,j}$. In algorithmic iterations $\mathbf{P}_{i,i}$ maybe used in place of $\mathbf{P}^i$. 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. The aggregated vector of local vectors $\mathbf{p}^i\in\mathbb{R}^{n^i}$ is $\mathbf{p}=(\mathbf{p}^1,\dots,\mathbf{p}^N)\in\mathbb{R}^d$, $d=\sum_{i=1}^N n^i$. We use $a\xrightarrow{k}b$ to indicate that robot a has taken relative measurement from robot b at timestep k.

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, we assume to be a unique integer belonging to the set $\mathcal{V} = \{1, \ldots, N\}$. 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

$$\mathbf{x}^i(k+1) = \mathbf{f}^i(\mathbf{x}^i(k), \mathbf{u}^i(k)) + \mathbf{g}^i(\mathbf{x}^i(k))\boldsymbol{\eta}^i(k), \quad k \in \mathbb{Z}_{\geq 0}, \tag{1}$$

where $\mathbf{x}^i \in \mathbb{R}^{n^i}$, $\mathbf{u}^i \in \mathbb{R}^{m^i}$, and $\boldsymbol{\eta}^i \in \mathbb{R}^{p^i}$ 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

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{g}(\mathbf{x}(k))\boldsymbol{\eta}(k), \quad k \in \mathbb{Z}_{\geq 0}, \tag{2}$$

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 $\mathbf{g}(\mathbf{x}) = \mathrm{Diag}(\mathbf{g}^1(\mathbf{x}^1), \cdots, \mathbf{g}^N(\mathbf{x}^N))$. The process noises $\boldsymbol{\eta}^i$, $i \in \mathcal{V}$, are independent zero-mean white Gaussian processes with a known positive definite variance $\mathbf{Q}^i(k) = \mathrm{E}[\boldsymbol{\eta}^i(k)\boldsymbol{\eta}^i(k)^\top]$. 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

$$\mathbf{z}_{i,j}(k) = \mathbf{h}_{i,j}(\mathbf{x}^i(k), \mathbf{x}^j(k)) + \boldsymbol{\nu}^i(k), \quad \mathbf{z}_{i,j} \in \mathbb{R}^{n_z^i}, \quad i \xrightarrow{k} j, \quad k \in \mathbb{Z}_{\geq 0}, \tag{3}$$

where $\mathbf{h}_{i,j}(\mathbf{x}^i, \mathbf{x}^j)$ 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 $\mathbf{R}^i(k) = \mathrm{E}[\boldsymbol{\nu}^i(k)\boldsymbol{\nu}^i(k)^\top]$. All noises are assumed to be white and mutually uncorrelated.

It is apparent that if the robots only rely on propagating their equations of motion (1) to obtain their location because of the noise in self-motion measurements, their estimates will grow unbounded. We show below how using an EKF, relative measurements between robots are used to improve the propagated states of the collective robotic team system. Here, we assume that all the sensor measurements are synchronized.

III. Split EKF, an alternative representation of Centralized EKF algorithm for CL

In this section, we review a new representation of the the centralized EKF CL algorithm. We refer to this alternative representation as the split EKF representation. The Split EKF representation enables our decentralized implementations discussed in the proceeding sections.

The centralized EKF CL algorithm is a straightforward application of EKF over the collective motion model (2) using the relative measurement model (3) [12]. That is, starting at $\hat{\mathbf{x}}^{i+}(0) \in \mathbb{R}^{n^i}$, $\mathbf{P}^{i+}(0) \in \mathbb{S}^{n^i}_{>0}$, $\mathbf{P}^{+}_{i,j}(0) = \mathbf{0}_{n^i \times n^j}$, for $i \in \mathcal{V}$ and $j \in \mathcal{V} \backslash \{i\}$, the propagation and update equation of EKF-based CL for $k \in \mathbb{Z}_{\geq 0}$ is given by

$$\hat{\mathbf{x}}^{-}(k+1) = \mathbf{f}(\hat{\mathbf{x}}^{+}(k), \mathbf{u}(k)), \tag{4a}$$

$$\mathbf{P}^{-}(k+1) = \mathbf{F}(k)\mathbf{P}^{+}(k)\mathbf{F}(k)^\top + \mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}(k)^\top, \tag{4b}$$

$$\hat{\mathbf{x}}^{+}(k+1) = \hat{\mathbf{x}}^{-}(k+1) + \mathbf{K}(k+1)\mathbf{r}^a(k+1), \tag{4c}$$

$$\mathbf{P}^{+}(k+1) = \mathbf{P}^{-}(k+1) - \mathbf{K}(k+1)\mathbf{S}_{a,b}\mathbf{K}(k+1)^\top. \tag{4d}$$

$$\mathbf{K}(k+1) = \begin{cases} \mathbf{0}, & \text{no relative measurement at } k+1, \\ \mathbf{P}^{-}(k+1)\mathbf{H}_{a,b}(k+1)^\top \mathbf{S}_{a,b}(k+1)^{-1}, & a \xrightarrow{k+1} b, \end{cases} \tag{4e}$$

where $\mathbf{F} = \mathrm{Diag}(\mathbf{F}^1, \cdots, \mathbf{F}^N)$, $\mathbf{G} = \mathrm{Diag}(\mathbf{G}^1, \cdots, \mathbf{G}^N)$ and $\mathbf{Q} = \mathrm{Diag}(\mathbf{Q}^1, \cdots, \mathbf{Q}^N)$, 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 $i \in \mathcal{V}$. Moreover, when a robot a takes a relative measurement from robot b at some given time $k+1$, the measurement residual and its covariance are, respectively,

$$\mathbf{r}^a(k+1) = \mathbf{z}_{a,b}(k+1) - \mathbf{h}_{a,b}(\hat{\mathbf{x}}^{a-}(k+1), \hat{\mathbf{x}}^{b-}(k+1)), \tag{5a}$$

$$\begin{aligned} \mathbf{S}_{a,b}(k+1) =& \mathbf{R}^a(k+1) + \mathbf{H}_{a,b}(k+1)\mathbf{P}^{-}(k+1)\mathbf{H}_{a,b}(k+1)^\top \\ =& \mathbf{R}^a(k+1) + \tilde{\mathbf{H}}_a(k+1)\mathbf{P}^{a-}(k+1)\tilde{\mathbf{H}}_a(k+1)^\top + \tilde{\mathbf{H}}_b(k+1)\mathbf{P}^{b-}(k+1)\tilde{\mathbf{H}}_b(k+1)^\top \\ &+ \tilde{\mathbf{H}}_b(k+1)\mathbf{P}^{-}_{ba}(k+1)\tilde{\mathbf{H}}_a(k+1)^\top + \tilde{\mathbf{H}}_a(k+1)\mathbf{P}^{-}_{a,b}(k+1)\tilde{\mathbf{H}}_b(k+1)^\top. \end{aligned} \tag{5b}$$

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

$$\mathbf{H}_{a,b}(k) = \begin{array}{c} \begin{array}{ccccccccc} 1 & \cdots & a & a+1 & \cdots & b & b+1 & \cdots & N \end{array} \\ \begin{bmatrix} \mathbf{0} & \cdots & \tilde{\mathbf{H}}_a(k) & \mathbf{0} & \cdots & \tilde{\mathbf{H}}_b(k) & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \end{array}, \tag{6}$$

$$\tilde{\mathbf{H}}_a(k) = \frac{\partial}{\partial \mathbf{x}^a}\mathbf{h}_{a,b}(\hat{\mathbf{x}}^{a-}(k), \hat{\mathbf{x}}^{b-}(k)), \quad \tilde{\mathbf{H}}_b(k) = \frac{\partial}{\partial \mathbf{x}^b}\mathbf{h}_{a,b}(\hat{\mathbf{x}}^{a-}(k), \hat{\mathbf{x}}^{b-}(k)).$$

It is worth recalling that, if the system model is linear, at any update incident at some timestep k, given the state update equation (4c), the Kalman gain (4e) minimizes the trace of $\mathbf{P}^{+}(k) = \mathrm{E}[\mathbf{e}(k)\mathbf{e}(k)^{\top}]$, where $\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{x}^{+}(k)$ (c.f. [19, page 146]).

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 component 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 follows, for $i \in \mathcal{V}$,

$$\hat{\mathbf{x}}^{i-}(k+1) = \mathbf{f}^i(\hat{\mathbf{x}}^{i+}(k), \mathbf{u}^i(k)), \tag{7a}$$

$$\mathbf{P}^{i-}(k+1) = \mathbf{F}^i(k)\mathbf{P}^{i+}(k)\mathbf{F}^i(k)^{\top} + \mathbf{G}^i(k)\mathbf{Q}^i(k)\mathbf{G}^i(k)^{\top}, \tag{7b}$$

$$\mathbf{P}^{-}_{i,j}(k+1) = \mathbf{F}^i(k)\mathbf{P}^{+}_{i,j}(k)\mathbf{F}^j(k)^{\top}, \quad j \in \mathcal{V}\backslash\{i\}, \tag{7c}$$

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i-}(k+1) + \mathbf{K}_i(k+1)\mathbf{r}^a(k+1), \tag{7d}$$

$$\mathbf{P}^{i+}(k+1) = \mathbf{P}^{i-}(k+1) - \mathbf{K}_i(k+1)\mathbf{S}_{a,b}(k+1)\mathbf{K}_i(k+1)^{\top}, \tag{7e}$$

$$\mathbf{P}^{+}_{i,j}(k+1) = \mathbf{P}^{-}_{i,j}(k+1) - \mathbf{K}_i(k+1)\mathbf{S}_{a,b}(k+1)\mathbf{K}_j(k+1)^{\top}, \quad j \in \mathcal{V}\backslash\{i\}, \tag{7f}$$

$$\mathbf{K}_i(k+1) = \begin{cases} \mathbf{0}, & \text{no relative measurement at } k+1, \\ (\mathbf{P}^{-}_{i,b}(k+1)\tilde{\mathbf{H}}_b^{\top} + \mathbf{P}^{-}_{i,a}(k+1)\tilde{\mathbf{H}}_a^{\top})\mathbf{S}_{a,b}{}^{-1}, & a \xrightarrow{k+1} b. \end{cases} \tag{7g}$$

Because $\mathbf{K}_i(k+1)\mathbf{S}_{a,b}(k+1)\mathbf{K}_i(k+1)^{\top}$ is a positive semi-definite term, the update equation (7e) clearly shows that relative measurement updates reduce the estimation uncertainty. Next, note that, because of the inherent coupling in cross-covariance terms (7c) and (7f), the EKF CL (7) can not be implemented in a decentralized manner, without all-to-all communication. Next, we review and offer a rigorous proof for an alternative representation of the centralized EKF CL algorithm which was originally proposed in [3] without proof. We refer to this alternative representation as split EKF CL. We use this alternative representation, in the proceeding section, to decouple the propagation stage of (7), and propose a decentralized implementation of the algorithm with reduced processing and communication cost per robot. The split EKF CL is presented in the following result.

Theorem III.1 (Split EKF CL, an alternative representation of EKF for CL)**.** *Consider the EKF CL algorithm in* (7) *with initial conditions* $\hat{\mathbf{x}}^{i+}(0) \in \mathbb{R}^{n^i}$, $\mathbf{P}^{i+}(0) \in \mathbb{S}^{n^i}_{>0}$, $\mathbf{P}^{+}_{i,j}(0) = \mathbf{0}_{n^i \times n^j}$, *for* $i \in \mathcal{V}$ *and* $j \in \mathcal{V}\backslash\{i\}$. *For* $i \in \mathcal{V}$, *let* $\mathbf{\Phi}^i(0) = \mathbf{I}_{n^i}$ *and* $\mathbf{\Pi}_{i,j}(0) = \mathbf{0}_{n^i \times n^j}$, $j \in \mathcal{V}\backslash\{i\}$. *Moreover, assume* *that* $\mathbf{F}^i(k)$, $i\in\mathcal{V}$, *is invertible at all* $k\in\mathbb{Z}_{\geq 0}$. *Next, for* $k\in\mathbb{Z}_{\geq 0}$, *let*

$$\boldsymbol{\Phi}^i(k+1)=\mathbf{F}^i(k)\boldsymbol{\Phi}^i(k),\quad i\in\mathcal{V}, \tag{8a}$$

$$\boldsymbol{\Gamma}_i(k+1)=\mathbf{0},\quad i\in\mathcal{V},\quad \textit{no relative measurement at } k+1, \tag{8b}$$

$$\boldsymbol{\Gamma}_a(k+1)=\left(\boldsymbol{\Pi}_{a,b}(k)\boldsymbol{\Phi}^b(k+1)^\top\tilde{\mathbf{H}}_b^\top+\boldsymbol{\Phi}^a(k+1)^{-1}\mathbf{P}^{a\text{-}}(k+1)\tilde{\mathbf{H}}_a^\top\right)\mathbf{S}_{a,b}^{-\frac{1}{2}},\quad a\xrightarrow{k+1}b, \tag{8c}$$

$$\boldsymbol{\Gamma}_b(k+1)=\left(\boldsymbol{\Phi}^b(k+1)^{-1}\mathbf{P}^{b\text{-}}(k+1)\tilde{\mathbf{H}}_b^\top+\boldsymbol{\Pi}_{ba}(k)\boldsymbol{\Phi}^a(k+1)^\top\tilde{\mathbf{H}}_a^\top\right)\mathbf{S}_{a,b}^{-\frac{1}{2}},\quad a\xrightarrow{k+1}b, \tag{8d}$$

$$\boldsymbol{\Gamma}_i(k+1)=(\boldsymbol{\Pi}_{i,b}(k)\boldsymbol{\Phi}^b(k+1)^\top\tilde{\mathbf{H}}_b^\top+\boldsymbol{\Pi}_{i,a}(k)\boldsymbol{\Phi}^a(k+1)^\top\tilde{\mathbf{H}}_a^\top)\,\mathbf{S}_{a,b}^{-\frac{1}{2}},\quad i\in\mathcal{V}\backslash\{a,b\},\quad a\xrightarrow{k+1}b, \tag{8e}$$

$$\boldsymbol{\Pi}_{i,j}(k+1)=\boldsymbol{\Pi}_{i,j}(k)+\boldsymbol{\Gamma}_i(k+1)\,\boldsymbol{\Gamma}_j(k+1)^\top,\quad i\in\mathcal{V},\quad j\in\mathcal{V}\backslash\{i\}. \tag{8f}$$

Then, we can rewrite (7c) *and* (7f), *respectively as*

$$\mathbf{P}^{\text{-}}_{i,j}(k+1)=\boldsymbol{\Phi}^i(k+1)\,\boldsymbol{\Pi}_{i,j}(k)\,\boldsymbol{\Phi}^j(k+1)^\top, \tag{9a}$$

$$\mathbf{P}^{+}_{i,j}(k+1)=\boldsymbol{\Phi}^i(k+1)\,\boldsymbol{\Pi}_{i,j}(k+1)\,\boldsymbol{\Phi}^j(k+1)^\top. \tag{9b}$$

Consequently, we can represent (7d) *and* (7e), *respectively, as*

$$\hat{\mathbf{x}}^{i+}(k+1)=\hat{\mathbf{x}}^{i\text{-}}(k+1)+\boldsymbol{\Phi}^i(k+1)\boldsymbol{\Gamma}_i(k+1)\bar{\mathbf{r}}^a(k+1), \tag{10a}$$

$$\mathbf{P}^{i+}(k+1)=\mathbf{P}^{i\text{-}}(k+1)-\boldsymbol{\Phi}^i(k+1)\boldsymbol{\Gamma}_i(k+1)\boldsymbol{\Gamma}_i^\top(k+1)\boldsymbol{\Phi}^i(k+1)^\top, \tag{10b}$$

where $\bar{\mathbf{r}}^a(k+1)=\mathbf{S}_{a,b}^{-\frac{1}{2}}\mathbf{r}^a(k+1)$, $i\in\mathcal{V}$ *and* $j\in\mathcal{V}\backslash\{i\}$.

Proof. First, we evaluate our statement regarding (9). Our proof is based on mathematical induction over $k\in\mathbb{Z}_{\geq 0}$; that is, we first verify (9) for $k=0$, then we assume our claim (9) holds for k, and evaluate it for $k+1$.

Let $k=0$. Then given (8) and the defined initial conditions, (9a) results in

$$\mathbf{P}^{\text{-}}_{i,j}(1)=\boldsymbol{\Phi}^i(1)\,\boldsymbol{\Pi}_{i,j}(0)\,\boldsymbol{\Phi}^j(1)^\top=\mathbf{F}^i(1)\,\mathbf{0}_{n^i\times n^j}\,\mathbf{F}^j(1)^\top=\mathbf{0}_{n^i\times n^j},$$

which matches exactly (7c) at $k=0$. Next, we consider the first step of induction on (9a). When there is no relative measurement at the first step, then given (8e)-(8d) and (8f), (9b) results in $\mathbf{P}^{+}_{i,j}(1)=\boldsymbol{\Phi}^i(1)\,\boldsymbol{\Pi}_{i,j}(1)\,\boldsymbol{\Phi}^j(1)^\top=\boldsymbol{\Phi}^i(1)\,\boldsymbol{\Pi}_{i,j}(0)\,\boldsymbol{\Phi}^j(1)^\top=\mathbf{0}_{n^i\times n^j}$, matching exactly the first step of (7f). However, when there is a robot a that takes relative measurement from robot b at the first step of the algorithm, then from (8e)-(8d), we have

$$\boldsymbol{\Gamma}_i(1)=(\boldsymbol{\Pi}_{i,b}(0)\boldsymbol{\Phi}^b(1)^\top\tilde{\mathbf{H}}_b^\top+\boldsymbol{\Pi}_{i,a}(0)\boldsymbol{\Phi}^a(1)^\top\tilde{\mathbf{H}}_a^\top)\,\mathbf{S}_{a,b}^{-\frac{1}{2}}=\mathbf{0},\quad i\in\mathcal{V}\backslash\{a,b\}$$

$$\boldsymbol{\Gamma}_a(1) = \left(\boldsymbol{\Pi}_{a,b}(0)\boldsymbol{\Phi}^b(1)^\top\tilde{\mathbf{H}}_b^\top + \boldsymbol{\Phi}^a(1)^{-1}\mathbf{P}^{a\text{-}}(1)\tilde{\mathbf{H}}_a^\top\right)\mathbf{S}_{a,b}^{-\frac{1}{2}} = \mathbf{F}^a(1)^{-1}\mathbf{P}^{a\text{-}}(1)\tilde{\mathbf{H}}_a^\top\mathbf{S}_{a,b}^{-\frac{1}{2}},$$

$$\boldsymbol{\Gamma}_b(1) = \left(\boldsymbol{\Phi}^b(1)^{-1}\mathbf{P}^{b\text{-}}(1)\tilde{\mathbf{H}}_b^\top + \boldsymbol{\Pi}_{b,a}(0)\boldsymbol{\Phi}^a(1)^\top\tilde{\mathbf{H}}_a^\top\right)\mathbf{S}_{a,b}^{-\frac{1}{2}} = \mathbf{F}^b(1)^{-1}\mathbf{P}^{b\text{-}}(1)\tilde{\mathbf{H}}_b^\top\mathbf{S}_{a,b}^{-\frac{1}{2}}.$$

Then, from (9b), we obtain

$$\mathbf{P}^{+}_{i,j}(1) = \mathbf{0}, \quad i \in \mathcal{V}\setminus\{a,b\},\ j \in \mathcal{V}\backslash\{i,a,b\},$$

$$\mathbf{P}^{+}_{a,b}(1) = \mathbf{P}^{+}_{b,a}(1)^\top = \boldsymbol{\Phi}^a(1)\boldsymbol{\Pi}_{a,b}(1)\boldsymbol{\Phi}^b(1)^\top = \mathbf{F}^a(1)(\boldsymbol{\Pi}_{a,b}(0) - \boldsymbol{\Gamma}_a(1)\,\boldsymbol{\Gamma}_b(1)^\top)\mathbf{F}^b(1)^\top$$

$$= -\mathbf{F}^a(1)\boldsymbol{\Gamma}_a(1)\,\boldsymbol{\Gamma}_b(1)^\top\mathbf{F}^b(1)^\top$$

$$= -\mathbf{F}^a(1)(\mathbf{F}^a(1)^{-1}\mathbf{P}^{a\text{-}}(1)\tilde{\mathbf{H}}_a^\top\mathbf{S}_{a,b}^{-\frac{1}{2}})(\mathbf{F}^b(1)^{-1}\mathbf{P}^{b\text{-}}(1)\tilde{\mathbf{H}}_b^\top\mathbf{S}_{a,b}^{-\frac{1}{2}})^\top\mathbf{F}^b(1)^\top$$

$$= -(\mathbf{P}^{a\text{-}}(1)\tilde{\mathbf{H}}_a^\top\mathbf{S}_{a,b}^{-1})\mathbf{S}_{a,b}(\mathbf{P}^{b\text{-}}(1)\tilde{\mathbf{H}}_b^\top\mathbf{S}_{a,b}^{-1})^\top,$$

which exactly matches (7f) as shown below (recall (7g)). In each case, (7f) reduces to

$$\mathbf{P}^{+}_{i,j}(1) = \mathbf{0}, \quad i \in \mathcal{V}\setminus\{a,b\},\ j \in \mathcal{V}\backslash\{i,a,b\},$$

$$\mathbf{P}^{+}_{a,b}(1) = \mathbf{P}^{+}_{b,a}(1)^\top = \mathbf{P}^{\text{-}}_{i,j}(k+1) - \mathbf{K}_i(k+1)\mathbf{S}_{a,b}(k+1)\mathbf{K}_j(k+1)^\top$$

$$= -(\mathbf{P}^{a\text{-}}(1)\tilde{\mathbf{H}}_a^\top\mathbf{S}_{a,b}^{-1})\mathbf{S}_{a,b}(\mathbf{P}^{b\text{-}}(1)\tilde{\mathbf{H}}_b^\top\mathbf{S}_{a,b}^{-1})^\top.$$

Assume now that the theorem statement holds for k. At time step $k+1$, first consider (9a). This implies

$$\mathbf{P}^{\text{-}}_{i,j}(k+1) = \boldsymbol{\Phi}^i(k+1)\boldsymbol{\Pi}_{i,j}(k)\boldsymbol{\Phi}^j(k+1)^\top$$

$$= \mathbf{F}^i(k+1)\boldsymbol{\Phi}^i(k)\boldsymbol{\Pi}_{i,j}(k)\boldsymbol{\Phi}^j(k)^\top\mathbf{F}^j(k+1)^\top = \mathbf{F}^i(k+1)\mathbf{P}^{+}_{i,j}(k)\mathbf{F}^j(k+1)^\top,$$

which confirms validity of (9a) at $k+1$. Next, consider (9b). When there is no relative measurement at $k+1$, we obtain from (9b) (recall (8e)-(8d) and (8f))

$$\mathbf{P}^{+}_{i,j}(k+1) = \boldsymbol{\Phi}^i(k+1)\boldsymbol{\Pi}_{i,j}(k+1)\boldsymbol{\Phi}^j(k+1)^\top = \boldsymbol{\Phi}^i(k+1)\boldsymbol{\Pi}_{i,j}(k)\boldsymbol{\Phi}^j(k+1)^\top = \mathbf{P}^{\text{-}}_{i,j}(k+1),$$

using (9a). Next, we evaluate (9b) when robot a takes a relative measurement from robot b at $k+1$. First notice that

$$\mathbf{K}_i(k+1) = \boldsymbol{\Phi}^i(k+1)\boldsymbol{\Gamma}_i(k+1)\mathbf{S}_{a,b}^{-\frac{1}{2}}, \quad i \in \mathcal{V}. \tag{11}$$

This follows from the following considerations

- for $i \in \mathcal{V}\backslash\{a,b\}$ (recall (8e), (9a),) we have

$$\boldsymbol{\Phi}^i(k+1)\boldsymbol{\Gamma}_i(k+1)\mathbf{S}_{a,b}^{-\frac{1}{2}} = \boldsymbol{\Phi}^i(k+1)(\boldsymbol{\Pi}_{i,b}(k)\boldsymbol{\Phi}^b(k+1)^\top\tilde{\mathbf{H}}_b^\top + \boldsymbol{\Pi}_{i,a}(k)\boldsymbol{\Phi}^a(k+1)^\top\tilde{\mathbf{H}}_a^\top)\,\mathbf{S}_{a,b}^{-1}$$

$$= (\mathbf{P}^{-}_{i,b}(k+1)\tilde{\mathbf{H}}_b^\top + \mathbf{P}^{-}_{i,a}(k+1)\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{a,b}^{-1} = \mathbf{K}_i(k+1);$$

- for $i=a$ (recall (8c), (9a),) we have

$$\begin{aligned}\mathbf{\Phi}^i(k+1)\mathbf{\Gamma}_i(k+1)\mathbf{S}_{a,b}^{-\frac{1}{2}} &= \mathbf{\Phi}^a(k+1)\big(\mathbf{\Pi}_{a,b}(k)\mathbf{\Phi}^b(k+1)^\top\tilde{\mathbf{H}}_b^\top + \mathbf{\Phi}^a(k+1)^{-1}\mathbf{P}^{a-}(k+1)\tilde{\mathbf{H}}_a^\top\big)\,\mathbf{S}_{a,b}^{-1}\\ &= (\mathbf{P}^{-}_{a,b}(k+1)\tilde{\mathbf{H}}_b^\top + \mathbf{P}^{a-}(k+1)\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{a,b}^{-1} = \mathbf{K}_a(k+1);\end{aligned}$$

- for $i=b$ (recall (8d), (9a),) we have

$$\begin{aligned}\mathbf{\Phi}^i(k+1)\mathbf{\Gamma}_i(k+1)\mathbf{S}_{a,b}^{-\frac{1}{2}} &= \mathbf{\Phi}^b(k+1)\big(\mathbf{\Phi}^b(k+1)^{-1}\mathbf{P}^{b-}(k+1)\tilde{\mathbf{H}}_b^\top + \mathbf{\Pi}_{b,a}(k)\mathbf{\Phi}^a(k+1)^\top\tilde{\mathbf{H}}_a^\top\big)\,\mathbf{S}_{a,b}^{-1}\\ &= (\mathbf{P}^{b-}(k+1)\tilde{\mathbf{H}}_b^\top + \mathbf{P}^{-}_{b,a}(k)\tilde{\mathbf{H}}_a^\top)\,\mathbf{S}_{a,b}^{-1} = \mathbf{K}_b(k+1).\end{aligned}$$

Therefore, we can rewrite (9b) (recall (8e)-(8d) and (8f)) as

$$\begin{aligned}\mathbf{P}^{+}_{i,j}(k+1) &= \mathbf{\Phi}^i(k+1)\mathbf{\Pi}_{i,j}(k+1)\mathbf{\Phi}^j(k+1)^\top = \mathbf{\Phi}^i(k+1)\mathbf{\Pi}_{i,j}(k)\mathbf{\Phi}^j(k+1)^\top -\\ &\qquad \mathbf{\Phi}^i(k+1)\mathbf{\Gamma}_i(k+1)\,\mathbf{\Gamma}_j(k+1)^\top\mathbf{\Phi}^j(k+1)^\top\\ &= \mathbf{P}^{-}_{i,j}(k+1) - \big(\mathbf{\Phi}^i(k+1)\mathbf{\Gamma}_i(k+1)\,\mathbf{S}_{a,b}^{-\frac{1}{2}}\big)\,\mathbf{S}_{a,b}\,\big(\mathbf{\Phi}^j(k+1)\mathbf{\Gamma}_j(k+1)\mathbf{S}_{a,b}^{-\frac{1}{2}})^\top\big)\\ &= \mathbf{P}^{-}_{i,j}(k+1) - \mathbf{K}_i(k+1)\mathbf{S}_{a,b}\mathbf{K}_j(k+1)^\top,\end{aligned}$$

which confirms validity of (9b) at $k+1$ when robot a takes relative measurement from robot b. This completes the proof of validity of (9b) for all $k \in \mathbb{Z}_{\geq 0}$. Subsequently, (10) follows, in a straightforward manner, from (11) now being valid for all $k \in \mathbb{Z}_{\geq 0}$. □

It is worth noticing that, using the alternative representation (9a) for the cross-covariance of the team members, the residual covariance $\mathbf{S}_{a,b}$ in (5b) can be expressed in an equivalent way as

$$\begin{aligned}\mathbf{S}_{a,b} =& \mathbf{R}^a(k+1) + \tilde{\mathbf{H}}_a\mathbf{P}^{a-}(k+1)\tilde{\mathbf{H}}_a^\top + \tilde{\mathbf{H}}_b\mathbf{P}^{b-}(k+1)\tilde{\mathbf{H}}_b^\top +\\ &\tilde{\mathbf{H}}_a\mathbf{\Phi}^a(k+1)\mathbf{\Pi}_{a,b}(k)\mathbf{\Phi}^b(k+1)^\top\tilde{\mathbf{H}}_b^\top + \tilde{\mathbf{H}}_b\mathbf{\Phi}^b(k+1)\mathbf{\Pi}_{b,a}(k)\mathbf{\Phi}^a(k+1)^\top\tilde{\mathbf{H}}_a^\top.\end{aligned} \tag{12}$$

For clarity of presentation, so far, we have assumed that there is only one relative measurement at each given timestep k. To process multiple synchronized measurements, we use *sequential updating* (c.f. e.g. [20, ch. 3],[21]). In the following, we obtain a compact representation for sequential updating procedure in split ELF CL, that will be used in the partial decentralization scheme of proceeding section.

Let $\mathcal{V}_{\mathrm{A}}(k)$ denote the set of the robots that have made an exteroceptive measurement at time k, $\mathcal{V}^i_{\mathrm{B}}(k)$ denote the landmark robots of robot $i \in \mathcal{V}_{\mathrm{A}}(k)$, and $\mathcal{V}_{\mathrm{A,B}}(k)$ represent the set of all landmark robots and the robots that have taken relative measurements. Then, when there are $n_s=\sum_{i=1}^{|\mathcal{V}_\mathrm{A}(k+1)|}|\mathcal{V}_\mathrm{B}^i(k+1)|$ multiple concurrent exteroceptive relative measurements at a timestep $k+1$, the sequential updating procedure prescribes the following update equation for the EKF CL algorithm in (7). Let $\hat{\mathbf{x}}^{i+}(k+1,0)=\hat{\mathbf{x}}^{i-}(k+1)$, $\mathbf{P}^{i+}(k+1,0)=\mathbf{P}^{i-}(k+1)$, $i\in\{1,\cdots,N\}$, and $\mathbf{P}_{i,l}^{+}(k+1,0)=\mathbf{P}_{i,l}^{i-}(k+1)$ for $l\in\{1,\cdots,N\}\backslash\{i\}$. The update at time $k+1$ is $\hat{\mathbf{x}}^{i+}(k+1)=\hat{\mathbf{x}}^{i+}(k+1,n_s)$, $\mathbf{P}^{i+}(k+1)=\mathbf{P}^{i+}(k+1,n_s)$, and $\mathbf{P}_{i,l}^{+}(k+1)=\mathbf{P}_{i,l}^{+}(k+1,n_s)$, $i\in\{1,\cdots,N\}$, $l\in\{1,\cdots,N\}\backslash\{i\}$, obtained from executing the following steps, starting at $j=1$,

for $a\in\mathcal{V}_\mathrm{A}(k+1)$,

for $b\in\mathcal{V}_\mathrm{B}^a(k+1)$,

$$\hat{\mathbf{x}}^{i+}(k+1,j)=\hat{\mathbf{x}}^{i-}(k+1,j-1)+\mathbf{K}_i(k+1,j)\mathbf{r}^a(k+1,j), \tag{13a}$$

$$\mathbf{P}^{i+}(k+1,j)=\mathbf{P}^{i-}(k+1,j-1)-\mathbf{K}_i(k+1)\mathbf{S}_{a,b}(k+1,j)\mathbf{K}_i(k+1,j)^\top, \tag{13b}$$

$$\mathbf{P}_{i,l}^{+}(k+1,j)=\mathbf{P}_{i,l}^{-}(k+1,j-1)-\mathbf{K}_i(k+1,j)\mathbf{S}_{a,b}(k+1,j)\mathbf{K}_j(k+1,j)^\top_,\; j\in\mathcal{V}\backslash\{i\}, \tag{13c}$$

$$j\leftarrow j+1,$$

where $\mathbf{r}^a(k+1,j)$, $\mathbf{S}_{a,b}(k+1,j)$, and $\mathbf{K}_i(k+1,j)$ are calculated, respectively, from (5a), (5b), and (7g) using $\hat{\mathbf{x}}^-(k+1)=\hat{\mathbf{x}}^+(k+1,j-1)$ and $\mathbf{P}^-(k+1)=\mathbf{P}^+(k+1,j-1)$.

By direct substitution, we can show that (13a), (13b) and (13c) can be represented in Split EKF CL form as follows

$$\hat{\mathbf{x}}^{i+}(k+1,j)=\hat{\mathbf{x}}^{i-}(k+1,j-1)+\mathbf{\Phi}^i(k+1)\mathbf{\Gamma}_i(k+1,j)\bar{\mathbf{r}}^a(k+1,j), \tag{14a}$$

$$\mathbf{P}^{i+}(k+1,j)=\mathbf{P}^{i-}(k+1,j-1)-\mathbf{\Phi}^i(k+1)\mathbf{\Gamma}_i(k+1,j)\mathbf{\Gamma}_i^\top(k+1,j)\mathbf{\Phi}^i(k+1)^\top, \tag{14b}$$

$$\mathbf{P}_{i,l}^{+}(k+1,j)=\mathbf{\Phi}^i(k+1)\,\mathbf{\Pi}_{i,l}(k+1,j-1)\,\mathbf{\Phi}^l(k+1)^\top_,\quad l\in\mathcal{V}\backslash\{i\}, \tag{14c}$$

where $\mathbf{\Pi}_{i,j}(k+1,0)=\mathbf{\Pi}_{i,j}(k)$, $\mathbf{\Pi}_{i,j}(k+1,j)=\mathbf{\Pi}_{i,j}(k+1,j-1)+\mathbf{\Gamma}_i(k+1,j)\,\mathbf{\Gamma}_j(k+1,j)^\top$. Here, $\mathbf{\Gamma}_i(k+1,j)$ is calculated from (8e)-(8d) wherein $\mathbf{S}_{a,b}$ at each j is calculated from (12) using $\hat{\mathbf{x}}^-(k+1)=\hat{\mathbf{x}}^+(k+1,j-1)$ and $\mathbf{P}^-(k+1)=\mathbf{P}^+(k+1,j-1)$. Consequently, $\bar{\mathbf{r}}^a(k+1,j)=\mathbf{S}_{a,b}(k+1,j)^{-\frac{1}{2}}\mathbf{r}^a(k+1,j)$. Notice that here, we can represent the final updated variables as

$$\hat{\mathbf{x}}^{i+}(k+1,n_s)=\hat{\mathbf{x}}^{i-}(k+1,0)+\mathbf{\Phi}^i(k+1)\sum\nolimits_{j=1}^{n_s}\mathbf{\Gamma}_i(k+1,j)\bar{\mathbf{r}}^a(k+1,j), \tag{15a}$$

$$\mathbf{P}^{i+}(k+1,n_s)=\mathbf{P}^{i-}(k+1,0)-\mathbf{\Phi}^i(k+1)\Big(\sum\nolimits_{j=1}^{n_s}\mathbf{\Gamma}_i(k+1,j)\mathbf{\Gamma}_i^\top(k+1,j)\Big)\mathbf{\Phi}^i(k+1)^\top, \tag{15b}$$

$$\mathbf{P}^{+}_{i,l}(k{+}1, n_s) = \mathbf{\Phi}^i(k+1)\Big(\sum\nolimits_{j=1}^{n_s} \mathbf{\Pi}_{i,l}(k+1, j-1)\Big)\mathbf{\Phi}^l(k+1)^\top. \tag{15c}$$

We use the compact representation (15) of the sequential updating procedure to develop a partially decentralized implementation which requires only one update message broadcast from the CCU (see Algorithm 2).

IV. Partially decentralized implementation of the EKF for CL

In this section, using the split EKF CL representation, introduced in Theorem III.1, we devise an implementation of the EKF for CL where the propagation stage is fully decentralized but the update gains are calculated and sent out to the robots in centralized manner.

Using split EKF CL representation, in [3] the authors proposed a fully decentralized implementation of the centralized EKF CL where each agent $i \in \mathcal{V}$ stores and evolves its local ($\hat{\mathbf{x}}^{i-}(k) \in \mathbb{R}^{n^i}$, $\mathbf{\Phi}^i(k) \in \mathbb{R}^{n^i \times n^i}$, $\mathbf{P}^{i-}(k{+}1) \in \mathbb{S}^{n^i}_{>0}$) along with a *local copy* of $\mathbf{\Pi}_{l,j}(k)$ of the entire team (i.e., each robot $i \in \mathcal{V}$ maintains $\mathbf{\Pi}^i_{l,j}(k)$ for $l \in \mathcal{V}$, $j \in \mathcal{V}\backslash\{l\}$). This way the propagation stage of the EKF CL algorithm is fully decentralized. Whenever there is a relative measurement in the team at some timestep $k+1$, say $a \xrightarrow{k+1} b$, robot a acquires ($\hat{\mathbf{x}}^{b-}(k{+}1) \in \mathbb{R}^{n^b}$, $\mathbf{\Phi}^b(k{+}1) \in \mathbb{R}^{n^b \times n^b}$, $\mathbf{P}^{b-}(k{+}1) \in \mathbb{S}^{n^b}_{>0}$) from the landmark robot b. Then, robot a is designated as the interim master which calculates and broadcasts the update message $\big(a \in \mathbb{R}$, $b \in \mathbb{R}$, $\bar{\mathbf{r}}^a \in \mathbb{R}^{n^a_z}$, $\mathbf{\Gamma}_a \in \mathbb{R}^{n^a \times n^a_z}$, $\mathbf{\Gamma}_b \in \mathbb{R}^{n^b \times n^a_z}$, ${\mathbf{\Phi}^b}^\top \tilde{\mathbf{H}}_b^\top {\mathbf{S}_{a,b}}^{-\frac{1}{2}} \in \mathbb{R}^{n^b \times n^a_z}$, ${\mathbf{\Phi}^a}^\top \tilde{\mathbf{H}}_a^\top {\mathbf{S}_{a,b}}^{-\frac{1}{2}} \in \mathbb{R}^{n^a \times n^a_z}\big)$ to the entire team. In this way, every agent $i \in \mathcal{V}$ is able to calculate a local copy of $\mathbf{\Gamma}^i_j$, $j \in \mathcal{V}$ to update its local state, error covariance matrix and its local copy of the $\mathbf{\Pi}^i_{l,j}(k)$. The algorithm in [3] 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. Also, notice that the success of the algorithm of [3] depends on maintaining perfect communication message deliveries in the team. Any incidence of message dropout at each agent will cause disparity between the local copy of $\mathbf{\Pi}_{l,j}(k)$'s at that agent and the local copies maintained by the rest of the team. As a result, message dropout jeopardizes the integrity of the proposed decentralized implementation.

Our goal here is to design an algorithm that imposes only $O(1)$ processing and storage cost per robot and also be robust to communication message dropouts. To accomplish these objectives, we propose that a CCU maintains the team cross-covariances, which is the source of high processing and storage costs. In the next section, we show that our proposed scheme is robust to message dropouts with the updated estimation of each robot receiving the update message being updated to be minimum variance at that given time.

Using the split EKF CL introduced in Theorem III.1, our proposed partially decentralized CL algorithm is as follows. Every robot $i \in \mathcal{V}$ maintains and propagates its propagated state estimate (7a) and its corresponding covariance matrix (7b), as well as, variable $\mathbf{\Phi}^i$ (8a). 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 $\mathbf{\Pi}_{i,j}$'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 ($\mathbf{z}_{a,b}$, $\hat{\mathbf{x}}^{a\text{-}}(k{+}1)$, $\mathbf{\Phi}^a(k{+}1)$, $\mathbf{P}^{a\text{-}}(k{+}1)$) from robot a and ($\hat{\mathbf{x}}^{b\text{-}}(k{+}1)$, $\mathbf{\Phi}^b(k{+}1)$, $\mathbf{P}^{b\text{-}}(k{+}1)$) from robot b. Then, using this information, which we refer to it as *landmark-message*, along with its locally maintained $\mathbf{\Pi}_{i,j}$'s, it calculates $\mathbf{r}_a$, $\mathbf{S}_{a,b}$ and $\mathbf{\Gamma}_i$, $i \in \mathcal{V}$, from respectively, (5a), (12) and (8e)-(8d). Then, the CCU sends to each robot $i \in \mathcal{V}$ its corresponding update message $(\bar{\mathbf{r}}^a, \mathbf{\Gamma}_i)$ so that the robot can update its local estimates using (10). It also updates its local $\mathbf{\Pi}_{i,j}$ using (8f), 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 1 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, i.e., $\mathcal{V}_{\text{missed}}(k+1)$, the set of agents missing the update message of the CCU at timestep $k+1$, is empty. This requirement is relaxed in the proceeding section, where we study the robustness of our proposed algorithm to message dropouts.

To include absolute measurements in Algorithm 1 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 $(\bar{\mathbf{r}}^a, \mathbf{\Gamma}_i)$ 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. [21, page 104] and references therein). Here, we assume

Assumption 1. *CCU has a pre-specified sequential-updating-order guideline, which indicates the priority order for implementing the measurement update.*

Algorithm 1 *Partially DCL*

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

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

The CCU initializes

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

Iteration k

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

$$\begin{aligned}
&\hat{\mathbf{x}}^{i\text{-}}(k+1)=\mathbf{f}^i(\hat{\mathbf{x}}^{i+}(k),\mathbf{u}^i(k)),\\
&\mathbf{\Phi}^i(k+1)=\mathbf{F}^i(k)\mathbf{\Phi}^i(k),\\
&\mathbf{P}^{i\text{-}}(k+1)=\mathbf{F}^i(k)\mathbf{P}^{i+}(k)\mathbf{F}^i(k)^\top+\mathbf{G}^i(k)\mathbf{Q}^i(k)\mathbf{G}^i(k)^\top.
\end{aligned}\tag{16}$$

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

$$\hat{\mathbf{x}}^{i+}(k+1)=\hat{\mathbf{x}}^{i\text{-}}(k+1),\quad \mathbf{P}^{i+}(k+1)=\mathbf{P}^{i\text{-}}(k+1),\tag{17}$$

and the CCU proceeds with

$$\mathbf{\Pi}_{i,j}(k+1)=\mathbf{\Pi}_{i,j}(k),\quad j\in\mathcal{V}\backslash\{i\}.\tag{18}$$

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,

$$\begin{aligned}
\textit{Landmark-message}^a&=\Big(\mathbf{z}_{a,b},\hat{\mathbf{x}}^{a\text{-}}(k+1),\mathbf{P}^{b\text{-}}(k+1),\mathbf{\Phi}^a(k+1)\Big),\\
\textit{Landmark-message}^b&=\Big(\hat{\mathbf{x}}^{b\text{-}}(k+1),\mathbf{P}^{b\text{-}}(k+1),\mathbf{\Phi}^b(k+1)\Big).
\end{aligned}\tag{19}$$

Given the *Landmark-message*, the CCU calculates

$$\mathbf{S}_{a,b}=\mathbf{R}^a+\tilde{\mathbf{H}}_a\mathbf{P}^{a\text{-}}\tilde{\mathbf{H}}_a^\top+\tilde{\mathbf{H}}_b^\top\mathbf{P}^{b\text{-}}\tilde{\mathbf{H}}_b+\tilde{\mathbf{H}}_a\mathbf{\Phi}^a\mathbf{\Pi}_{a,b}\mathbf{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top+\tilde{\mathbf{H}}_b\mathbf{\Phi}^b\mathbf{\Pi}_{ba}\mathbf{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top,\tag{20}$$

as well as $\mathbf{r}^a$ and $\mathbf{\Gamma}_i$'s using (5a), (8e)-(8d), respectively. It obtains $\bar{\mathbf{r}}^a=(\mathbf{S}_{a,b})^{-\frac{1}{2}}\mathbf{r}^a$ and then passes the following data to every robot $i\in\mathcal{V}$ in the network:

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

Every robot $i\in\mathcal{V}\backslash\mathcal{V}_{\text{missed}}(k+1)$, upon receiving its respective *update-message*^{i}, updates its state estimate and the corresponding covariance

$$\hat{\mathbf{x}}^{i+}(k+1)=\hat{\mathbf{x}}^{i\text{-}}(k+1)+\mathbf{\Phi}^i(k+1)\textit{update-message}^i(2)\ \textit{update-message}^i(1),\tag{21a}$$

$$\mathbf{P}^{i+}(k+1)=\mathbf{P}^{i\text{-}}(k+1)-\mathbf{\Phi}^i(k+1)\textit{update-message}^i(2)\ \textit{update-message}^i(2)^\top\mathbf{\Phi}^i(k+1)^\top.\tag{21b}$$

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

$$\mathbf{\Pi}_{i,j}(k+1)=\mathbf{\Pi}_{i,j}(k)-\mathbf{\Gamma}_i\mathbf{\Gamma}_j^\top,\quad \text{if } (i,j)\notin\mathcal{V}_{\text{missed}}(k+1)\times\mathcal{V}_{\text{missed}}(k+1).\tag{22}$$

where $\mathcal{V}_{\text{missed}}(k+1)$ is the set of agents missing the update message of the CCU at timestep $k+1$.

3: $k\leftarrow k+1$

To implement sequential updating procedure, the robots making measurements inform the CCU and indicate to CCU what their landmark robots are. Therefore, the CCU knows $\mathcal{V}_{\text{A}}(k+1)$ and $\mathcal{V}^i_{\text{B}}(k+1)$'s, and sorts both of these sets according to it's sequential-updating-order guide-

Algorithm 2 CCU's sequential updating procedure for multiple in-network measurement at time $k+1$

Require: Initialization ($j=0$): CCU obtains the following information from each robot $a \in \mathcal{V}_{\text{A}}(k+1)$ and all of its landmarks $b \in \mathcal{V}_{\text{B}}^{a}(k+1)$,

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

The CCU initializes the following variables

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

Iteration j: CCU proceeds with the following calculations.

1: **for** $a \in \mathcal{V}_{\text{A}}(k+1)$ **do**

2: **for** $b \in \mathcal{V}_{\text{B}}^{a}(k+1)$ **do**

3: CCU calculates $\tilde{\mathbf{H}}_a$, $\tilde{\mathbf{H}}_b$ and $\mathbf{r}^a$ using $\hat{\mathbf{x}}^{\text{-}a}(k+1) = \hat{\mathbf{x}}^{+a}(k+1,j-1)$ and $\hat{\mathbf{x}}^{\text{-}b}(k+1) = \hat{\mathbf{x}}^{+b}(k+1,j-1)$. Then, using these measurement matrices and $\hat{\mathbf{P}}^{\text{-}a}(k+1) = \hat{\mathbf{P}}^{+a}(k+1,j-1)$, $\hat{\mathbf{P}}^{\text{-}b}(k+1) = \hat{\mathbf{P}}^{+b}(k+1,j-1)$ and $\mathbf{\Pi}_{a,b}(k) = \mathbf{\Pi}_{a,b}(k+1,j-1)$, CCU calculates $\mathbf{S}_{a,b}$ from (12) and subsequently $\bar{\mathbf{r}}^{a}(k+1,j) = (\mathbf{S}_{a,b}(k+1,j))^{-\frac{1}{2}}\mathbf{r}^{a}(k+1,j)$ and $\mathbf{\Gamma}_i(k+1,j)$ from (8e)-(8d) for $i \in \mathcal{V}$. Next, CCU updates the state and the covariance of all the robots in $i \in \mathcal{V}_{\text{A,B}}(k+1)$ as follows

$$\hat{\mathbf{x}}^{+i}(k+1,j) = \hat{\mathbf{x}}^{+i}(k+1,j) + \mathbf{\Phi}^{i}(k+1)\,\mathbf{\Gamma}_i(k+1,j)\,\bar{\mathbf{r}}^{a}(k+1,j), \tag{23a}$$
$$\mathbf{P}^{+i}(k+1,j) = \mathbf{P}^{+i}(k+1,j) - \mathbf{\Phi}^{i}(k+1)\mathbf{\Gamma}_i(k+1,j)\mathbf{\Gamma}_i(k+1,j)^{\top}\mathbf{\Phi}^{i}(k+1)^{\top}. \tag{23b}$$

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

$$\mathbf{\Pi}_{i,l}(k+1,j) = \mathbf{\Pi}_{i,l}(k+1,j-1) - \mathbf{\Gamma}_i(k+1,j)\mathbf{\Gamma}_l(k+1,j)^{\top}, \quad \text{if } (i,l) \notin \mathcal{V}_{\text{missed}}(k+1) \times \mathcal{V}_{\text{missed}}(k+1).$$

4: $j \leftarrow j+1$

5: **end for**

6: **end for**

7: CCU sets $\mathbf{\Pi}_{i,l}(k+1) = \mathbf{\Pi}_{i,l}(k+1,n_s)$, where $n_s = \sum_{a \in \mathcal{V}_{\text{A}}(k+1)} |\mathcal{V}_{\text{B}}^{a}(k+1)|$.

8: CCU broadcasts the following update messages for robot $i \in \mathcal{V}$

$$\textit{update-message}^{i} = \big(\textstyle\sum_{j=1}^{n_s}(\mathbf{\Gamma}_i(k+1,j)\bar{\mathbf{r}}^{a}(k+1,j)), \sum_{j=1}^{n_s}(\mathbf{\Gamma}_i(k+1,j)\mathbf{\Gamma}_i(k+1,j)^{\top})\big). \tag{24}$$

line. Then, the CCU according to its sequential-updating-order can collect, one at a time, the *Landmark-message* of the robots in $\mathcal{V}_{\text{A}}(k+1)$ and process and the measurement according to equations (19)-(22) of Algorithm 1. After the first robot in $\mathcal{V}_{\text{A}}(k+1)$, the next robots use their updated local estimate and error covariance to generate their *Landmark-message*.

An alternative implementation for sequential updating where the CCU issues only one collective update message is possible using the compact representation of the sequential updating equations (14) and (15). Algorithm 2 describes the details of such an implementation that we use to modify Algorithm 1 to accommodate multiple concurrent measurement processing. Here, the CCU first collects all the landmark messages (19) of the robots in $\mathcal{V}_{\text{A}}(k+1)$ and $\mathcal{V}_{\text{A}}(k+1)$.

Then, the CCU computes the collective update message. Note that in this implementation, the CCU should create a local copy of the state estimate and the error covariance equations of the robots in $\mathcal{V}^{-}_{\text{A,B}}(k+1)$ (see. (23)), because these updates are needed to compute $\mathbf{S}_{a,b}$ and other intermediate variables. An alternative implementation is also possible where the update message for every robot $i \in \mathcal{V}_{\text{A,B}}(k+1)$ is

$$\textit{update-message}^i =$$
$$\big((\mathbf{\Phi}^i)^{-1}(\hat{\mathbf{x}}^{+i}(k+1,n_s) - \hat{\mathbf{x}}^{-i}(k+1)), -(\mathbf{\Phi}^i)^{-1}(\mathbf{P}^{+i}(k+1,n_s) - \mathbf{P}^{-i}(k+1))(\mathbf{\Phi}^i)^{-T}\big).$$

instead of (24). This is because the CCU already has computed the update state estimates and the corresponding covariances of robot $i \in \mathcal{V}_{\text{A,B}}(k+1)$ as part of partial updating procedure, i.e, $\hat{\mathbf{x}}^{+i}(k+1) = \hat{\mathbf{x}}^{+i}(k+1,n_s)$, and $\mathbf{P}^{+i}(k+1) = \mathbf{P}^{+i}(k+1,n_s)$.

Finally, observe that our partially decentralized algorithm is robust to permanent team member dropouts. The CCU only suffers from a processing and communication cost until it can confirm that the dropout is permanent. In the next section, we study the robustness of the proposed partially DCL Algorithm 1 to occasional in-network message dropouts.

V. Accounting for in-network message dropouts

In this section, we study the robustness of Algorithm 1 against the occasional communication link failures between robots and the CCU. We devise a modification that maintains the desired minimum variance update property of the state estimate of the robots receiving the update message, when the robot dynamics are linear. For nonlinear robot dynamics, the results will be suboptimal due to the linearization approximation.

Our guarantees are based on the assumption 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 at some update times, we do not update the estimate 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.

Consider a centralized CL where we always are able to update the state estimate equations of the robots involved in a relative measurement. Next, without loss of generality, assume that we do not update the state estimate of robots $\{m+1,\cdots,N\}$, $2 < m < N+1$ using the relative measurement taken by robot $a \notin \{m+1, \cdots, N\}$ from robot $b \notin \{m+1, \cdots, N\}$ at some time $k+1$, (this is equivalent to assuming in the partially decentralized operation we have $\mathcal{V}_{\text{missed}}(k+1) = \{m+1, \cdots, N\}$). 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 (7a)-(7c). The following result gives the minimum variance update equation for robots $\{1, \cdots, m\}$, when the robotic team model is linear.

Theorem V.1 (Partial updating)**.** *Let the robotic team equations of motion and the measurement models be linear. Consider a centralized EKF based CL where a FC uses the relative measurement taken by robot $a \notin \mathcal{V}_{missed}(k+1)$ from robot $b \notin \mathcal{V}_{missed}(k+1)$ at some time $k+1 > 0$ to only update the states of robots $\mathcal{V} \backslash \mathcal{V}_{missed}(k+1) = \{1, \cdots, m\}$, i.e.,*

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i-}(k+1) + \mathbf{K}_i(k+1)\mathbf{r}^a(k+1), \quad i \in \mathcal{V} \backslash \mathcal{V}_{missed}(k+1) \tag{25a}$$

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i-}(k+1) \quad i \in \mathcal{V}_{missed}(k+1). \tag{25b}$$

Let $\mathbf{K}_{1:m} = [\mathbf{K}_1^\top, \cdots, \mathbf{K}_m^\top]^\top$. Then, the Kalman gain $\mathbf{K}_{1:m}$ that minimizes the trace of $\mathbf{P}^+(k+1)$ (the minimum variance partial state update gain) is given by

$$\mathbf{K}_i = (\mathbf{P}^-_{i,b}(k+1)\tilde{\mathbf{H}}_b^\top + \mathbf{P}^-_{i,a}(k+1)\tilde{\mathbf{H}}_a^\top)\,\mathbf{S}_{a,b}{}^{-1}, \quad i \in \mathcal{V} \backslash \mathcal{V}_{missed}(k+1). \tag{26}$$

Moreover, the corresponding team covariance update is given by

$$\mathbf{P}^{i+}(k+1) = \begin{cases} \mathbf{P}^{i-}(k+1), & i \in \mathcal{V}_{missed}(k+1), \\ \mathbf{P}^{i-}(k+1) - \mathbf{K}_i\mathbf{S}_{a,b}(k+1)\mathbf{K}_i(k+1)^\top & \textit{otherwise} \end{cases} \tag{27}$$

For cross-covariances we obtain

$$\mathbf{P}^+_{i,j}(k+1) = \begin{cases} \mathbf{P}^-_{i,j}(k+1), & (i,j) \in \mathcal{V}_{missed}(k+1) \times \mathcal{V}_{missed}(k+1), \\ \mathbf{P}^-_{i,j}(k+1) - \mathbf{K}_i(k+1)\mathbf{S}_{a,b}(k+1)\mathbf{K}_j(k+1)^\top & \textit{otherwise} \end{cases} \tag{28}$$

where we defined and used the pseudo *gain*

$$\mathbf{K}_i = (\mathbf{P}^-_{i,b}(k+1)\tilde{\mathbf{H}}_b^\top + \mathbf{P}^-_{i,a}(k+1)\tilde{\mathbf{H}}_a^\top)\,\mathbf{S}_{a,b}{}^{-1}, \quad i \in \mathcal{V}_{missed}(k+1). \tag{29}$$

Proof. We can obtain Kalman gain $\mathbf{K}_{1:m}$ that minimizes the trace of $\mathbf{P}^+(k+1)$ from $\partial \text{Tr}(\mathbf{P}^+(k+1))/\partial \mathbf{K}_{1:m} = \mathbf{0}$. Let $\hat{\mathbf{x}}^+_{1:m} = (\hat{\mathbf{x}}^{1+}, \cdots, \hat{\mathbf{x}}^{m+})$, $\hat{\mathbf{x}}^+_{m+1:N} = (\hat{\mathbf{x}}^{m+1+}, \cdots, \hat{\mathbf{x}}^{N+})$. Next, we obtain

$\mathrm{Tr}(\mathbf{P}^{+}(k+1))$. Given (25), when the system and measurement models are linear, the state error update at $k+1$ is given by

$$\begin{bmatrix} \mathbf{e}^{+}_{1:m}(k+1) \\ \mathbf{e}^{+}_{m+1:N}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{1:m}(k+1) - \mathbf{x}^{+}_{1:m}(k+1) \\ \mathbf{x}_{m+1:N}(k+1) - \mathbf{x}^{+}_{m+1:N}(k+1) \end{bmatrix} =$$

$$\begin{bmatrix} -\mathbf{K}_{1:m} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{\nu}^{a}(k+1) \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} (\mathbf{I}_m - \mathbf{K}_{1:m}\bar{\mathbf{H}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N-m} \end{bmatrix}\begin{bmatrix} \mathbf{x}_{1:m}(k+1) - \mathbf{x}^{-}_{1:m}(k+1) \\ \mathbf{x}_{m+1:N}(k+1) - \mathbf{x}^{-}_{m+1:N}(k+1) \end{bmatrix},$$

where we used $\bar{\mathbf{H}} = \begin{array}{c} \begin{array}{ccccccccc} 1 & \cdots & a & a+1 & \cdots & b & b+1 & \cdots & m \end{array} \\ \begin{bmatrix} \mathbf{0} & \cdots & \tilde{\mathbf{H}}_a(k+1) & \mathbf{0} & \cdots & \tilde{\mathbf{H}}_b(k+1) & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \end{array}$. Recall that $\mathbf{P}^{+}(k+1) = \mathrm{E}[\mathbf{e}^{+}(k+1)\mathbf{e}^{+}(k+1)^{\top}]$ which is equal to

$$\mathbf{P}^{+}(k+1) = \left[\begin{array}{c|c} \mathbf{P}^{+}_{1:m,1:m}(k+1) & \mathbf{P}^{+}_{1:m,m+1:N}(k+1) \\ \hline \mathbf{P}^{+}_{1:m,m+1:N}(k+1)^{\top} & \mathbf{P}^{+}_{m+1:N,m+1:N}(k+1) \end{array}\right] = \begin{bmatrix} \mathbf{K}_{1:m}\mathbf{R}_a\mathbf{K}^{\top}_{1:m} & \mathbf{0}_{m\times(N-m)} \\ \mathbf{0}_{(N-m)\times m} & \mathbf{0}_{(N-m)\times(N-m)} \end{bmatrix}$$
$$+ \begin{bmatrix} (\mathbf{I}_m - \mathbf{K}_{1:m}\bar{\mathbf{H}}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N-m} \end{bmatrix} \left[\begin{array}{c|c} \mathbf{P}^{-}_{1:m,1:m}(k+1) & \mathbf{P}^{-}_{1:m,m+1:N}(k+1) \\ \hline \mathbf{P}^{-}_{m+1:N,1:m}(k+1) & \mathbf{P}^{-}_{m+1:N,m+1:N}(k+1) \end{array}\right] \begin{bmatrix} (\mathbf{I}_m - \mathbf{K}_{1:m}\bar{\mathbf{H}})^{\top} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N-m} \end{bmatrix}. \tag{30}$$

Then, we have

$$\begin{aligned} \mathrm{Tr}(\mathbf{P}^{+}(k+1)) =& \,\mathrm{Tr}(\mathbf{K}_{1:m}\mathbf{R}_a\mathbf{K}^{\top}_{1:m}) + \mathrm{Tr}((\mathbf{I}_m - \mathbf{K}_{1:m}\bar{\mathbf{H}})\mathbf{P}^{-}_{1:m,1:m}(k+1)(\mathbf{I}_m - \mathbf{K}_{1:m}\bar{\mathbf{H}})^{\top}) \\ &+ \mathrm{Tr}(\mathbf{P}^{-}_{m+1:N,m+1:N}(k+1)) \\ =& \,\mathrm{Tr}(\mathbf{P}^{-}_{1:m,1:m}(k+1)) - 2\,\mathrm{Tr}(\mathbf{K}_{1:m}\bar{\mathbf{H}}\mathbf{P}^{-}_{1:m,1:m}(k+1)) + \\ &\mathrm{Tr}(\mathbf{K}_{1:m}(\mathbf{R}_a + \bar{\mathbf{H}}\mathbf{P}^{-}_{1:m,1:m}(k+1)\bar{\mathbf{H}}^{\top})\mathbf{K}^{\top}_{1:m}) + \mathrm{Tr}(\mathbf{P}^{-}_{m+1:N,m+1:N}(k+1)). \end{aligned}$$

As a result,

$$\begin{aligned} \partial\mathrm{Tr}(\mathbf{P}^{+}(k+1))/\partial\mathbf{K}_{1:m} =& -2\,\mathbf{P}^{-}_{1:m,1:m}(k+1)\bar{\mathbf{H}}^{\top} + 2\,(\mathbf{R}_a + \bar{\mathbf{H}}\mathbf{P}^{-}_{1:m,1:m}(k+1)\bar{\mathbf{H}}^{\top})\mathbf{K}^{\top}_{1:m} \\ =& -2\,\mathbf{P}^{-}_{1:m,1:m}(k+1)\bar{\mathbf{H}}^{\top} + 2\,\mathbf{S}_{a,b}\mathbf{K}^{\top}_{1:m}. \end{aligned}$$

Therefore, the gain $\mathbf{K}_{1:m}$ that minimizes the trace of $\mathbf{P}^{+}_{1:m}(k+1)$ is $\mathbf{K}_{1:m} = \bar{\mathbf{H}}\mathbf{P}^{-}_{1:m,1:m}(k+1)\mathbf{S}^{-1}_{a,b}$, which equivalently can be extended in robot-wise components to give us (26). For the covariance update, from (30), we obtain

$$\begin{aligned} \mathbf{P}^{+}_{1:m,1:m}(k+1) =& \,(\mathbf{I}_m - \mathbf{K}_{1:m}\bar{\mathbf{H}})\mathbf{P}^{-}_{1:m,1:m}(k+1)(\mathbf{I}_m - \mathbf{K}_{1:m}\bar{\mathbf{H}})^{\top} + \mathbf{K}_{1:m}\mathbf{R}_a\mathbf{K}^{\top}_{1:m} \\ =& \,\mathbf{P}^{-}_{1:m,1:m}(k+1) - \mathbf{K}_{1:m}\mathbf{S}_{a,b}\mathbf{K}^{\top}_{1:m}, \end{aligned} \tag{31a}$$

$$\mathbf{P}^{+}_{m+1:N,m+1:N}(k+1) = \mathbf{P}^{-}_{m+1:N,m+1:N}(k+1), \tag{31b}$$

$$\mathbf{P}^{+}_{1:m,m+1:N}(k+1) = (\mathbf{I}_m - \mathbf{K}_{1:m}\bar{\mathbf{H}})\mathbf{P}^{-}_{1:m,m+1:N}(k+1) = \tag{31c}$$

$$\left(\mathbf{I}_m - \begin{bmatrix}\mathbf{K}_1\\ \vdots \\ \mathbf{K}_m\end{bmatrix} \begin{array}{c} \begin{matrix} 1 & \cdots & a & a+1 & \cdots & b & b+1 & \cdots & m \end{matrix} \\ \begin{bmatrix} \mathbf{0} & \cdots & \tilde{\mathbf{H}}_a(k) & \mathbf{0} & \cdots & \tilde{\mathbf{H}}_b(k) & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix}\end{array}\right)\mathbf{P}^{-}_{1:m,m+1:N}(k+1) =$$

$$\left(\mathbf{I}_m - \begin{bmatrix}\mathbf{K}_1\mathbf{S}_{a,b}\mathbf{S}^{-1}_{a,b}\\ \vdots \\ \mathbf{K}_m\mathbf{S}_{a,b}\mathbf{S}^{-1}_{a,b}\end{bmatrix} \begin{array}{c} \begin{matrix} 1 & \cdots & a & a+1 & \cdots & b & b+1 & \cdots & m \end{matrix} \\ \begin{bmatrix} \mathbf{0} & \cdots & \tilde{\mathbf{H}}_a(k) & \mathbf{0} & \cdots & \tilde{\mathbf{H}}_b(k) & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix}\end{array}\right)\mathbf{P}^{-}_{1:m,m+1:N}(k+1),$$

where

$$\mathbf{P}^{-}_{1:m,m+1:N}(k+1) = \begin{bmatrix} \mathbf{P}^{-}_{1,m+1}(k+1) & \cdots & \mathbf{P}^{-}_{1,N}(k+1) \\ \vdots & \cdots & \vdots \\ \mathbf{P}^{-}_{a,m+1}(k+1) & \cdots & \mathbf{P}^{-}_{a,N}(k+1) \\ \vdots & \cdots & \vdots \\ \mathbf{P}^{-}_{b,m+1}(k+1) & \cdots & \mathbf{P}^{-}_{b,N}(k+1) \\ \vdots & \cdots & \vdots \\ \mathbf{P}^{-}_{m,m+1}(k+1) & \cdots & \mathbf{P}^{-}_{m,N}(k+1) \end{bmatrix}.$$

Recalling the definition of the pseudo-gains (29), then (31) results in (27) and (28). □

When the robot and/or measurement models of the robotic team are nonlinear, the guarantees provided in Theorem (V.1) are only suboptimal due to the linearization approximation.

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 $i \in \mathcal{V}\backslash\{1, \cdots, m\}$ and also the cross-covariance update terms using the pseudo gain $\mathbf{K}_{m+1:N}$ stay the same. As such, the decomposition technique of split EKF CL used to develop the partially decentralized algorithm of Section IV is valid here. Thus, we can implement exactly Algorithm 1 as is, while the robots missing the update message of the CCU do not update their estimates and the CCU does not update the $\mathbf{\Pi}_{i,j}$ when $(i, j) \in \mathcal{V}_{\text{missed}}(k+1) \times \mathcal{V}_{\text{missed}}(k+1)$. Therefore, this algorithm is robust to message dropouts and the estimates of the robots receiving the update message, as stated above, are minimum variance.

VI. Comparative performance evaluations in simulations

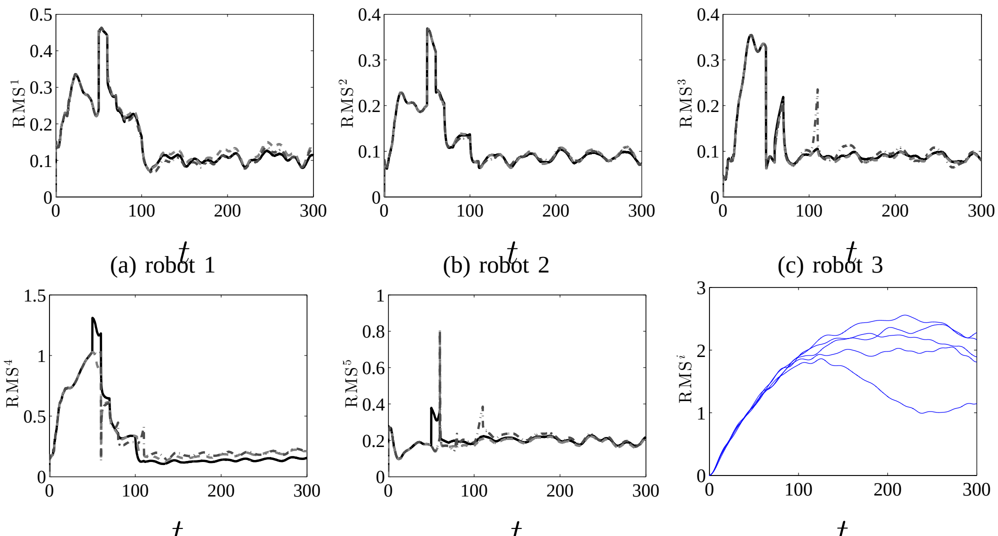
We compare the performance of the proposed partially DCL algorithm with and without occasional communication failure in simulations. We use a team of five robots moving on a flat terrain of $25\text{m}\times 25\text{m}$ area with constant linear velocity of 0.25 m/s and the rotational velocity drawn 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; $\mathbf{z}^i = [x^i, y^i]^\top + \boldsymbol{\nu}_z^i$ 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 $RMS^i(k) = \sqrt{\frac{1}{M}\sum_{j=1}^{M} \mathbf{e}_j^i(k)^\top \mathbf{e}_j^i(k)}$. Figure 1 shows the results for the measurement and communication scenarios explained in Table I.

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

VII. Conclusions

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



(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.

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 DCL algorithm is robust to communication link failures between some robots and the CCU and the updated estimates of the robots receiving the CCU's update message are 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 [22] when the communication link is restored between the corresponding robot and the CCU.

ACKNOWLEDGMENT

The authors would like to thank Stephen Rounds for early conversations.

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. S. Kia, S. Rounds, and S. Martínez, “Cooperative localization for mobile agents: a recursive decentralized algorithm based on Kalman filter decoupling,” *IEEE Control Systems Magazine*, vol. 36, no. 2, pp. 86–101, 2016.

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

[5] 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.

[6] 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.

[7] P. O. Arambel, C. Rago, and R. K. Mehra, “Covariance intersection algorithm for distributed spacecraft state estimation,” in *American Control Conference*, (Arlington, VA), pp. 4398–4403, 2001.

[8] 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*, (Tokyo, Japan), pp. 1412–1417, 2013.

[9] H. Li and F. Nashashibi, “Cooperative multi-vehicle localization using split covariance intersection filter,” *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 2, pp. 33–44, 2013.

[10] D. Marinescu, N. O’Hara, and V. Cahill, “Data incest in cooperative localisation with the common past-invariant ensemble kalman filter,” (Istanbul, Turkey), pp. 68–76, 2013.

[11] 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.

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

[13] S. S. Kia, S. Rounds, and S. Martínez, “A centralized-equivalent decentralized implementation of extended Kalman filters for cooperative localization,” in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, (Chicago, IL), pp. 3761–3765, September 2014.

[14] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, “Recursive decentralized collaborative localization for sparsely communicating robots,” in *Proceedings of Robotics: Science and Systems*, (AnnArbor, Michigan), June 2016.

[15] 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.

[16] 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.

[17] S. E. Webster, J. M. Walls, L. L. Whitcomb, and R. M. Eustice, “Decentralized extended information filter for single-beacon cooperative acoustic navigation: Theory and experiments,” *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 957–974, 2013.

[18] S. S. Kia, S. Rounds, and S. Martínez, “Cooperative Localization under message dropouts via a partially decentralized EKF scheme,” in *IEEE Int. Conf. on Robotics and Automation*, (Seattle, WA, USA), pp. 5977–5982, May 2015.

[19] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*. Chapman & Hall/CRC, 2 ed., 2011.

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

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

[22] 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.