Cooperative Localization for Mobile Agents

A recursive decentralized algorithm based on Kalman-filter decoupling

Solmaz S. Kia, Stephen Rounds and Sonia Martínez

Technological advances in ad-hoc networking and miniaturization of electro-mechanical systems are making possible the use of large numbers of mobile agents (for example, mobile robots, human agents, unmanned vehicles) to perform surveillance, search and rescue, transport and delivery tasks in aerial, underwater, space, and land environments. However, the successful execution of such tasks often hinges upon accurate position information, which is needed in lower-level locomotion and path-planning algorithms. Common techniques for localization of mobile robots are the classical preinstalled beacon-based localization algorithms [1], fixed feature-based simultaneous localization and mapping (SLAM) algorithms [2], and GPS navigation [3]; see Figure 1 for further details. However, these localization techniques work based on assumptions such as the existence of distinct and static features that can be revisited often, or line-of-sight to GPS satellites, which may not be feasible for operations such as search and rescue [4], [5], environment monitoring [6], [7], and oceanic exploration [8]. In case of GPS navigation, there is also a current concern about signal jamming for outdoor navigation, especially for UAV coordination and control. Instead, cooperative localization (CL) is emerging as an alternative localization technique that can be employed in such scenarios.

In CL, a group of mobile agents with processing and communication capabilities use relative measurements with respect to each other (no reliance on external features) as a feedback signal to *jointly estimate* the poses (both position and orientation) of all team members, which results in an increased accuracy for the entire team. The principle behind CL is similar to that of differential GPS, by which inaccurate GPS measurements are corrected with fixed range measurements from stations with known locations. However, CL goes beyond in its application since no fixed stations or landmarks, and continuous access to GPS measurements (even if they are inaccurate) are assumed.

The particular appeal of CL relies on the fact that sporadic access to accurate localization information by a particular robot results in a net benefit for the rest of the team. This is possible thanks to the coupling that is created through the state estimation process. Moreover, the use of other robots as landmarks also eliminates the potential problems on data association and perception, because the robots can be equipped with distinctive features or markers that are easily recognizable by other robots. Another nice feature of CL is its cost effectiveness, since it does not require extra hardware beyond the operational components normally used in cooperative robotic tasks. In such situations, agents are normally equipped with unique identifiers and sensors which, enable them to identify and locate other group members. To achieve coordination, these agents often broadcast their status information to one another. In addition, given affordability and wide availability of communication devices, CL has also emerged as an augmentation system to compensate for poor odometric measurements, and/or noisy and distorted measurements from other sensor suites such as onboard IMU systems; see, for example, [9].

The idea of exploiting relative robot-to-robot measurements for localization can be traced back to [10], where members of a mobile robotic team were divided into two groups, which took turns remaining stationary as landmarks for the others. Later, [11] removed the necessity of some robots to be stationary, and also introduced the term cooperative localization to refer to this localization technique. Since then, many cooperative localization algorithms using various estimation strategies such as extended Kalman filters (EKF) [12], maximum likelihood [13], maximum a posteriori (MAP) [14], and particle filters [15], [16], [17] have been developed. CL techniques to handle system and measurement models with non-Gaussian noises are also discussed in [18], [19].

Although, due to its independence from environmental features or GPS information, CL is a very attractive concept for multi-robot localization, it comes with new challenges associated with its implementation with acceptable communication, memory, and processing costs. CL is a joint estimation process that results in highly coupled pose estimation for the entire robotic team. These couplings/cross-correlations are created due to the relative measurement updates. Accounting for these coupling/cross-correlations is crucial for both filter consistency and also for propagating the benefit of a robot-to-robot measurement update to the entire group. In "Cooperative localization via EKF" these features are highlighted through detailed examination of equations of a centralized EKF CL, and a simulation demonstration.

A centralized implementation of CL is the most straightforward mechanism to keep an accurate account of the couplings and, as a result, to obtain more accurate solutions. In a centralized scheme, *at every time step*, a single device, either a leader robot or a fusion center (FC), gathers and processes information from the entire team. Then, it broadcasts back the estimated location results to each robot (see, for example, [12], [20]). But, such a central operation is highly energy consuming since it incurs a high processing cost on the FC and a high communication cost on both FC and each robotic team member. Moreover, central operations lack robustness due to existence of single failure point.This energy inefficiency and lack of robustness make the centralized implementation less preferable.

A major challenge in developing decentralized CL (DCL) algorithms is how to maintain a precise account of cross-correlations and couplings between the agents' estimates without invoking *all-to-all* communication at each time step. The design and analysis of DCL algorithms that maintain the consistency of the estimation process while maintaining 'reasonable' communication and computation costs have been the subject of extensive research since the conception of the CL idea. "Decentralized cooperative localization: how to account for intrinsic correlations in cooperative localization," gives an overview of some of the DCL algorithms in the literature, with a special focus on how these algorithms maintain/account for intrinsic correlations of CL strategy. Subsequently, "The interim master DCL algorithm: a tightly coupled DCL strategy based on Kalman-filter decoupling," provides readers with an example of a DCL algorithm that maintains the exact account of cross covariances between robotic team member pose estimates without resorting to all-to-all communication. A preliminary version of this example DCL strategy appeared in [21]. The reader interested on technical analysis and details beyond decentralization for CL can find a brief literature guide in "Further Reading."

Notations: Before proceeding further, the notations are introduced. $\mathbb{M}_n$, $\mathbf{0}_{n\times m}$ (when $m=1$, $\mathbf{0}_n$ is used), and $\mathbf{I}_n$, respectively, denote the set of real positive definite matrices of dimension $n\times n$, the zero matrix of dimension $n\times m$, and the identity matrix of dimension $n\times n$. $\mathbf{A}^\top$ represents the transpose of matrix $\mathbf{A}\in\mathbb{R}^{n\times m}$. The block diagonal matrix of set of matrices $\mathbf{A}_1,\dots,\mathbf{A}_N$ is $\text{Diag}(\mathbf{A}_1,\cdots,\mathbf{A}_N)$. 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. For a finite set V its cardinality is represented by $|V|$. In a team of N agents, the local variables associated with agent i are distinguished by the superscript i, for example, $\mathbf{x}^i$ is the state of agent i, $\hat{\mathbf{x}}^i$ is its state estimate, and $\mathbf{P}^i$ is the covariance matrix of its state estimate. In this paper, the term *cross covariance* refers to the correlation terms between two agents in the covariance matrix of the entire network. The cross covariance of the state vectors of agents i and j is $\mathbf{P}_{ij}$. The propagated and updated variables, say $\hat{\mathbf{x}}^i$, at time step k are denoted by $\hat{\mathbf{x}}^{i-}(k)$ and $\hat{\mathbf{x}}^{i+}(k)$, respectively. The time step argument of the variables as well as matrix dimensions are dropped whenever they are clear from the context. In a network of N agents, $\mathbf{p}=(\mathbf{p}^1,\dots,\mathbf{p}^N)\in\mathbb{R}^d$, $d=\sum_{i=1}^N n^i$ is the aggregated vector of local vectors $\mathbf{p}^i\in\mathbb{R}^{n^i}$.

Cooperative localization via EKF

This section provides an overview of a CL strategy that employs an EKF following [22]. By a close examination of this algorithm, it is possible to explain why accounting for the intrinsic cross-correlations in CL is both crucial for filter consistency and a key to spreading the benefit of an update of a relative robot-to-robot measurement to the entire team. This section also discusses the communication cost of implementing this algorithm in a centralized manner.

Consider a group of N mobile agents with communication, processing, and measurement capabilities. Depending on the adopted CL algorithm, communication can be in (a) a bidirectional manner with a FC, (b) a single broadcast to the entire team or (c) multihop fashion of every agent rebroadcasting every received message intended to reach the entire team; see Figure 2. Each agent has a detectable unique identifier (UID), which, without loss of generality, is assumed to be a unique integer belonging to the set $\mathcal{V} = \{1, \dots, N\}$. Using a set of proprioceptive sensors, every agent $i \in \mathcal{V}$ measures its self-motion, for example, by compass readings and/or wheel encoders, and uses it to propagate its equations of motion

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

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 state vector, the input vector, and the process noise vector of agent i. Here, $\mathbf{f}^i(\mathbf{x}^i, \mathbf{u}^i)$ and $\mathbf{g}^i(\mathbf{x}^i)$, are, respectively, the system function and process noise coefficient function of the agent $i \in \mathcal{V}$. The state vector of each agent can be composed of variables that describe the robot's global pose in the world (for example, latitude, longitude, direction), as well as other variables potentially needed to model the robots dynamics (for example, steering angle, actuation dynamics). The team can consist of heterogeneous agents, nevertheless, the collective motion equation of the team can be represented by

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{g}(\mathbf{x}(k))\boldsymbol{\eta}(k), \tag{2}$$

where, $\mathbf{x} = (\mathbf{x}^1, \cdots, \mathbf{x}^N)$, $\mathbf{u} = (\mathbf{u}^1, \cdots, \mathbf{u}^N)$, $\boldsymbol{\eta} = (\boldsymbol{\eta}^1, \cdots, \boldsymbol{\eta}^N)$, $\mathbf{f}(\mathbf{x}, \mathbf{u}) = (\mathbf{f}^1(\mathbf{x}^1, \mathbf{u}^1), \cdots, \mathbf{f}^N(\mathbf{x}^N, \mathbf{u}^N))$, and $\mathbf{g}(\mathbf{x}) = \mathrm{Diag}(\mathbf{g}^1(\mathbf{x}^1), \cdots, \mathbf{g}^N(\mathbf{x}^N))$.

If each agent only relies on propagating its equation of motion in (1) using self-motion measurements, the state estimate error drifts due to the noise term $\boldsymbol{\eta}^i(k)$, with a standard deviation that grows unbounded with time. To reduce the growth rate of this estimation error, a CL strategy can be employed. Thus, let every agent $i \in \mathcal{V}$ also carry exteroceptive sensors to monitor the environment to detect, uniquely, the other agents $j \in \mathcal{V}$ in the team and take relative measurements

$$\mathbf{z}_{ij}(k+1) = \mathbf{h}_{ij}(\mathbf{x}^i(k), \mathbf{x}^j(k)) + \boldsymbol{\nu}^i(k), \tag{3}$$

where $\mathbf{z}_{ij} \in \mathbb{R}^{n_z^i}$ from them, for example, relative pose, relative range, relative bearing measurements, or both. Here, $\mathbf{h}_{ij}(\mathbf{x}^i, \mathbf{x}^j)$ is the measurement model and $\boldsymbol{\nu}^i$ is the measurement noise of agent $i \in \mathcal{V}$. Relative-measurement feedback, as shown below, can help the agents improve their localization accuracy, though the overall uncertainty can not be bounded; see [22]. The tracking performance can be improved significantly if agents have occasional absolute positioning information, for example, via GPS or relative measurements taken from a fixed landmark with a priori known absolute location. Any absolute pose measurement by an agent $i \in \mathcal{V}$, for example, through intermittent GPS access, is modeled by $\mathbf{z}_{ii}(k+1) = \mathbf{h}_{ii}(\mathbf{x}^i(k)) + \bar{\boldsymbol{\nu}}^i(k)$. The agents can obtain concurrent exteroceptive absolute and relative measurements.

In the technical developments below, all the process noises $\boldsymbol{\eta}^i$ and the measurement noise $\boldsymbol{\nu}^i$, $i \in \mathcal{V}$, are assumed to be independent zero-mean white Gaussian processes with, respectively, known positive definite variances $\mathbf{Q}^i(k) = E[\boldsymbol{\eta}^i(k)\boldsymbol{\eta}^i(k)^\top]$, $\mathbf{R}^i(k) = E[\boldsymbol{\nu}^i(k)\boldsymbol{\nu}^i(k)^\top]$, and $\bar{\mathbf{R}}^i(k) = E[\bar{\boldsymbol{\nu}}^i(k)\bar{\boldsymbol{\nu}}^i(k)^\top]$. Moreover, let all the sensor noises be white and mutually uncorrelated and all sensor measurements be synchronized. Then, the centralized EKF CL algorithm is a straightforward application of EKF over the collective motion model of the robotic team (2) and measurement model (3). The propagation stage of this algorithm is

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

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

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, for all $i \in \mathcal{V}$, $\mathbf{F}^i = \frac{\partial}{\partial \mathbf{x}^i}\mathbf{f}(\hat{\mathbf{x}}^{i+}(k), \mathbf{u}^i(k))$, and $\mathbf{G}^i = \frac{\partial}{\partial \mathbf{x}^i}\mathbf{g}(\hat{\mathbf{x}}^{i+}(k))$.

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

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

$$\mathbf{S}_{ab} = \mathbf{H}_{ab}(k+1)\mathbf{P}^{-}(k+1)\mathbf{H}_{ab}(k+1)^{\top} + \mathbf{R}^a(k+1). \tag{5b}$$

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

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

An absolute measurement by a robot $a \in \mathcal{V}$ can be processed similarly, except that in (6), $\tilde{\mathbf{H}}_b$ becomes zero, while in (5), the index b should be replaced by a and $\mathbf{R}^a(k+1)$ should be replaced by $\bar{\mathbf{R}}^a(k+1)$. Then, the Kalman filter gain is given by

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

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

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

$$\mathbf{P}^{+}(k+1) = \mathbf{P}^{-}(k+1) - \mathbf{K}(k+1)\mathbf{S}_{ab}\mathbf{K}(k+1)^{\top}. \tag{7b}$$

Because $\mathbf{K}(k+1)\mathbf{S}_{ab}\mathbf{K}(k+1)^{\top}$ is a positive semidefinite term, the update equation (7b) shows that any relative measurement update results in a reduction of the estimation uncertainty.

To explore the relationship among the estimation equations of each robot, the aforementioned collective form of the EKF CL is expressed in terms of its agent-wise components in Algorithm 1. Here, the Kalman filter gain is partitioned into $\mathbf{K} = \begin{bmatrix}\mathbf{K}_1^{\top}, \cdots, \mathbf{K}_N^{\top}\end{bmatrix}^{\top}$, where $\mathbf{K}_i \in \mathbb{R}^{n^i \times n_z^i}$ is the portion of the Kalman-gain used to update the pose estimate of the agent $i \in \mathcal{V}$. To process multiple synchronized measurements, *sequential updating* (consult, for example, [23, Ch. 3],[24]) is employed.

Algorithm 1 showcases the role of past correlations in a CL strategy. First, observe that, despite having decoupled equations of motion, the source of the coupling in the propagation phase is the cross covariance equation (16c). Upon incidence of a relative measurement between agents a and b, the cross covariance term between robots a and b becomes nonzero, and its evolution in time requires the information of these two agents. Thus, these two agents have to either communicate with each other all the time or a central operator has to take over the propagation stage. As the number of relative measurements grow, more nonzero cross covariance terms are created. As a result, the communication cost to perform the propagation grows, requiring the data exchange all the time with either a FC or all-to-all agent communications, even when there is no relative measurement in the network. The update equations (18) are also coupled and their calculations need, in principle, a FC. The next observation regarding the role of the cross covariance terms can be deduced from studying the Kalman-gain equation (19). As this equation shows, when an agent a takes a relative measurement from agent b, any agent whose pose estimate is correlated with either of agents a and b in the past, (that is, $\mathbf{P}^{-}_{ib}(k+1)$ and/or $\mathbf{P}^{-}_{ia}(k+1)$ are nonzero) has a nonzero Kalman-gain and, as a result, the agent benefits from this measurement update. The same is true in the case of an absolute measurement taken by a robot a.

The following simple simulation study demonstrates the significance of maintaining an accurate account of cross covariance terms between the state estimates of the team members. This simulation study involves a team of 3 mobile robots moving on a flat terrain whose equations of motion in a fixed reference frame, for $i \in \{1,2,3\}$, are modeled as

$$
\begin{aligned}
x^i(k+1) =& x^i(k) + V^i(k)\cos(\phi(k))\,\delta t,\\
y^i(k+1) =& y^i(k) + V^i(k)\sin(\phi(k))\,\delta t,\\
\phi^i(k+1) =& \phi^i(k) + \omega(k)\,\delta t,
\end{aligned}
$$

where $V^i(k)$ and $\omega^i(k)$ are true linear and rotational velocities of robot i at time k and δt is the stepsize. Here, the pose vector of each robot is $\mathbf{x}^i = [x^i,\, y^i,\, \phi^i]^\top$. Every robot uses odometric sensors to measure its linear $V^i_m(k) = V^i(k) + \eta^i_V(k)$ and rotational $\omega^i_m(k) = \omega^i(k) + \eta^i_\omega(k)$, velocities, where η^i_V and η^i_ω are their respective contaminating measurement noise. The standard deviation of $\eta^i_V(k)$, $i \in \{1,2,3\}$, is $0.1V^i(k)$, while the standard deviation of η^i_ω is $1\,\mathrm{deg}/s$, for robot 1 and robot 2, and $0.5\,\mathrm{deg}/s$ for robot 3. Robots $\{1,2,3\}$ can take relative pose measurements from one another. This relative measurements are corrupted by additive

Gaussian measurement noise with standard deviations of, respectively, $(0.05\,\text{m}, 0.05\,\text{m}, 1\,\text{deg}/s)$, $(0.05\,\text{m}, 0.05\,\text{m}, 2\,\text{deg}/s)$, $(0.07\,\text{m}, 0.07\,\text{m}, 1.5\,\text{deg}/s)$ for robots 1, 2 and 3. Here, it is also assumed that robot 1 can obtain absolute position measurement with a standard deviation of $(0.1\text{m}, 0.1\text{m})$ for the measurement noise. Figure 3 demonstrates the x-coordinate estimation error (solid line) and the 3σ error bound (dashed lines) of these robots when they (a) only propagate their equations of motion using self-motion measurements (black plots), (b) employ an EKF CL ignoring past correlations between the estimates of the robots (blue plots), (c) employ an EKF CL with an accurate account of past correlations (red plots). As this figure shows, employing a CL strategy improves the localization accuracy by reducing both the estimation error and its uncertainty. However, as plots in blue show, ignoring the past correlations (here cross covariances) among the robots state estimates results in overly optimistic estimations (notice the almost vanished 3σ error bound in blue plots while the solid blue line goes out of these bounds, an indication of inconsistent estimates). In contrast, by taking into account the past correlations, more consistent estimates are obtained (see red plots).

Figure 3 also showcases the role of past cross covariances in spreading the benefit of a relative measurement between two robots, or of an absolute measurement by a robot to the rest of the team. For example, consider robot 2. In the time interval $[10, 90]$ seconds, robot 1 is taking a relative measurement from robot 2. As a result, the state estimation equation of robot 1 and robot 2 are correlated, that is, the cross covariance term between these two robots is nonzero. Therefore, in the time interval $[90, 110]$ seconds, when the estimation update is due to the relative measurement taken by robot 3 from robot 1, the state estimate of robot 2 is also improved (see red plots). In the time interval $[190, 240]$ seconds, when the estimation update is due to the absolute measurement taken by robot 1, both robot 2, and 3 also benefit from this measurement update due to past correlations (see the red plots). Figure 4 shows the trajectories of the robots when they apply EKF CL strategy. For more enlightening simulation studies, the interested reader is referred to [22].

Decentralized cooperative localization: how to account for intrinsic correlations in cooperative localization

Given that

(a) past correlations cannot be ignored,

(b) they are useful to increase the localization accuracy of the team,

(c) the coupling that the correlations create in the state estimation equations of team members is the main challenge in developing a decentralized cooperative localization algorithm,

regardless of the technique, two distinct trends can be observed in the literature for the design of DCL algorithms: "loosely coupled" strategies, and "tightly coupled" strategies; see Figure 5.

In the loosely coupled DCL methodology, only one or both of the agents involved in a relative measurement update their estimates using that measurement. Here, an exact account of the "network" of correlations (see Figure 5) due to the past relative measurement updates is not maintained. However, to ensure estimation consistency, some steps are taken to fix this problem. Examples of loosely coupled DCL are given in [8], [25], [26], [27] and [28]. In the algorithm of [8], only the agent obtaining the relative measurement updates its state. Here, to produce consistent estimates, a bank of EKFs is maintained at each agent. Using an accurate book keeping of the identity of the agents involved in previous updates and the age of such information, each of these filters is only updated when its propagated state is not correlated to the state involved in the current update equation. Although this technique does not impose a particular communication graph on the network, the computational complexity, the large memory demand, and the growing size of information needed at each update time are its main drawbacks. In [25], it is assumed that the relative measurements are in the form of relative pose. This enables the agent taking the relative measurement to use this measurement and its current pose estimate to obtain and broadcast a pose and the associated error covariance of its landmark agent (the landmark agent is the agent the relative measurement is taken from). Then, the landmark agent uses the covariance intersection method (see [29], [30]) to fuse the newly acquired pose estimate with its own current estimate to increase its estimation accuracy. Covariance intersection for DCL is also used in [26] for the localization of a group of space vehicles communicating over a fixed ring topology. Here, each vehicle propagates a model of the equation of motion of the entire team and, at the time of relative pose measurements, the vehicle fuses its estimate of the team states and of its landmark vehicle using covariance intersection. Another example of the use of split covariance intersection is given in [27] for intelligent transportation vehicles localization. [28] uses a common past-invariant ensemble Kalman pose estimation filter in another loosely-coupled CL approach for intelligent vehicles. This algorithm is very similar to the decentralized covariance intersection data fusion method described above, with the main difference that it operates with ensembles instead of with means and covariances. Even though the covariance intersection method produces consistent estimates for a loosely coupled DCL strategy, this method is known to produce overly conservative estimates. Overall, the loosely coupled algorithms have the advantage of not imposing any particular connectivity condition on the team. However, they are conservative by nature, because they do not enable other agents in the network to fully benefit from measurement updates.

In the tightly coupled DCL methodology, the goal is to exploit the 'network' of correlations created across the team (see Figure 5), so that the benefit of the update can be extended beyond the agents involved in a given relative measurement. However, this advantage comes at a potentially higher computational, storage, and/or communication cost. The dominant trend in developing decentralized cooperative localization algorithms in this way is to distribute the computation of components of a centralized algorithm among team members. Some of the examples for this class of DCL are given in [31], [22], [14], [32], [33]. In a straightforward fashion, decentralization can be conducted as a multicentralized CL, wherein each agent broadcasts its own information to the entire team. Then, every agent can calculate and reproduce the centralized pose estimates acting as a FC [31]. Besides a high processing cost for each agent, this scheme requires all-to-all agent communication at the time of each information exchange. A DCL algorithm distributing computations of an EKF centralized CL algorithm is proposed in [22]. To decentralize the cross covariance propagation, [22] uses a singular-value decomposition to split each cross covariance term between the corresponding two agents. Then, each agent propagates its portion. However, at update times, the separated parts must be combined, requiring an all-to-all agent communication in the correction step. Another DCL algorithm based on decoupling the propagation stage of an EKF CL using new intermediate variables is proposed in [21]. But here, unlike [22], 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, [14] and [33] present DCL strategies using MAP estimation procedure. In [14], computations of a centralized MAP CL algorithm are distributed among all the team members. In [33], the amount of data required to be passed between mobile agents to obtain the benefits of cooperative trajectory estimation locally is reduced by letting each agent to treat the others as moving beacons whose estimate of positions is only required at communication/measurement times. The aforementioned techniques all assume that communication messages are delivered, as prescribed, perfectly, all the time. A DCL approach equivalent to a centralized CL, when possible, which handles both limited communication ranges and time-varying communication graphs, is proposed in [32]. This technique uses an information transfer scheme wherein each agent broadcasts all its locally available information to every agent within its communication radius at each time step. The broadcasted information of each agent includes the past and present measurements, as well as past measurements previously received from other agents. The main drawback of this method is its high communication and memory cost, which may not be affordable in applications with limited communication bandwidth and storage resources.

The interim master DCL algorithm: a tightly coupled DCL strategy based on Kalman filter decoupling

Because of its recursive and simple structure, the EKF is a very popular estimation strategy. However, as discussed in "Cooperative localization via EKF," a naive decentralized implementation of EKF requires an all-to-all communication at every time step of the algorithm. This section describes how by exploiting a special pattern in the propagation estimation equations, tightly coupled *exact* decentralized implementations of EKF for CL with reduced communication workload per agent can be obtained. Here, "exact" means that if these decentralized implementations are initialized the same as a centralized EKF, then, they produce the same state estimate and the associated state error covariance of the centralized filter. The focus in this section is on the algorithm of [22], and a detailed treatment of the interim master DCL (IMDCL) algorithm of [21].

The algorithm of [22] and the IMDCL algorithm are developed based on the observation that, in localization problems, normally, the interest is in the *explicit* value of the pose estimate and the error covariance associated with it, while cross covariance terms are only required in the update equations. Such an observation promoted the proposal of an *implicit* tracking of cross covariance terms by splitting them into intermediate variables that can be propagated locally by the agents. Then, cross covariance terms can be recovered by putting together these intermediate variables at any update incidence. Let the last measurement update be in time step k and assume that for m subsequent and consecutive steps no relative measurement incidence takes place among the team members, that is, no intermediate measurement update is conducted in this time interval. In such a scenario, the propagated cross covariance terms for these m consecutive steps are given by

$$\mathbf{P}^{-}_{ij}(k+l) = \mathbf{F}^{i}(k+l-1)\,\cdots\,\mathbf{F}^{i}(k)\,\mathbf{P}^{+}_{ij}(k)\,\mathbf{F}^{j}(k)^{\top}\,\cdots\,\mathbf{F}^{j}(k+l-1)^{\top}, \quad l \in \{1,\cdots,m\}, \tag{8}$$

for $i \in \mathcal{V}$ and $j \in \mathcal{V}\backslash\{i\}$. That is, at each time step after k, the propagated cross covariance term is obtained by recursively multiplying its previous value by the Jacobian of the system function of agent i on the left and by the transpose of the Jacobian of the system function of agent j at that time step on the right. Based on this observation, [22] decomposed the last updated cross covariance term $\mathbf{P}^{+}_{ij}(k)$ between any agent i and any other agent j of the team into two parts (for example, using the singular-value decomposition technique). Then, agent i propagated the left portion, and agent j propagated the right portion. Note that, as long as there is no relative measurement among team members, each agent can propagate its portion of the cross covariance term locally without needing to communication with others. Such a decomposition in [22] led to a fully decentralized estimation algorithm during the propagation cycle. However, in the update stage, all the agents needed to communicate with one anther to put together the split cross covariance terms and proceed with the update stage. The approach to obtain IMDCL, which is outlined below, is also based on using intermediate variables to decouple the cross covariance propagation equations. However, this alternative decomposition allows every agent to update its pose estimate and its associated covariance in a centralized equivalent manner, using merely a scalable communication message that is received from the team member that takes the relative measurement. As such, the IMDCL algorithm removes the necessity of an all-to-all communication in the update stage and replaces it with propagating a constant size communication message that holds the crucial piece of information needed in the update stage.

In particular, the IMDCL algorithm is developed based on the observation that $\mathbf{P}^{-}_{ij}(k+l)$ in (8) is composed of the following 3 parts: (a) $\mathbf{F}^{i}(k+l-1)\cdots\mathbf{F}^{i}(k)$, which is local to agent i, (b) the $\mathbf{P}^{+}_{ij}(k)$, which does not change unless there is relative measurement among the team members, and (c) $\mathbf{F}^{j}(k)^{\top}\cdots\mathbf{F}^{j}(k+l-1)^{\top}$, which is local to agent j. Motivated by how these three parts evolve in time, the propagated cross covariances (16c) are written as

$$\mathbf{P}^{-}_{ij}(k+1) = \mathbf{\Phi}^{i}(k+1)\mathbf{\Pi}_{ij}(k)\mathbf{\Phi}^{j}(k+1)^{\top}, \quad k \in \{0,1,2,\cdots\}, \tag{9}$$

where, for $i \in \mathcal{V}$,

$$\mathbf{\Phi}^{i}(0) = \mathbf{I}_{n^i}, \quad \mathbf{\Phi}^{i}(k+1) = \mathbf{F}^{i}(k)\mathbf{\Phi}^{i}(k), \quad k \in \{0,1,2,\cdots\}. \tag{10}$$

It is interesting to notice the resemblance of (10) and the transition matrix for discrete-time systems. The variable $\mathbf{\Pi}_{ij} \in \mathbb{R}^{n^i \times n^j}$, for $i,j \in \mathcal{V}$ and $i \neq j$ in (9) is also time-varying and initialized at $\mathbf{\Pi}_{ij}(0) = \mathbf{0}_{n^i \times n^j}$. When there is no relative measurement at time $k+1$, (9) results into $\mathbf{\Pi}_{ij}(k+1) = \mathbf{\Pi}_{ij}(k)$. However, when there is a relative measurement among the team members, $\mathbf{\Pi}_{ij}$ must be updated in a manner such that $\mathbf{P}^{-}_{ij}(k{+}2) = \mathbf{\Phi}^{i}(k{+}2)\mathbf{\Pi}_{ij}(k{+}1)\mathbf{\Phi}^{j}(k{+}2)^{\top}$. To this end, notice that the update equations (17) and (19) of the centralized CL algorithm can be rewritten by replacing the cross covariance terms by (9) (recall that in the update stage, the assumption is that robot a has taken measurement from robot b)

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

and the Kalman-gain is

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

where

$$\mathbf{\Gamma}_{i} = (\mathbf{\Pi}_{ib}(k)\mathbf{\Phi}^{b^{\top}}\tilde{\mathbf{H}}_{b}^{\top} - \mathbf{\Pi}_{ia}(k)\mathbf{\Phi}^{a^{\top}}\tilde{\mathbf{H}}_{a}^{\top})\,\mathbf{S}_{ab}^{-\frac{1}{2}}, \; i \in \mathcal{V}\backslash\{a,b\}, \tag{12a}$$

$$\mathbf{\Gamma}_{a} = (\mathbf{\Pi}_{ab}(k)\mathbf{\Phi}^{b^{\top}}\tilde{\mathbf{H}}_{b}^{\top} - (\mathbf{\Phi}^{a})^{-1}\mathbf{P}^{a-}\tilde{\mathbf{H}}_{a}^{\top})\,\mathbf{S}_{ab}^{-\frac{1}{2}}, \tag{12b}$$

$$\mathbf{\Gamma}_{b} = ((\mathbf{\Phi}^{b})^{-1}\mathbf{P}^{b-}\tilde{\mathbf{H}}_{b}^{\top} - \mathbf{\Pi}_{ba}(k)\mathbf{\Phi}^{a^{\top}}\tilde{\mathbf{H}}_{a}^{\top})\,\mathbf{S}_{ab}^{-\frac{1}{2}}. \tag{12c}$$

Generally, $\mathbf{F}^i(k)$ is invertible for all $k \geq 0$ and $i \in \mathcal{V}$. Therefore, $\mathbf{\Phi}^i(k)$, for all $k \geq 0$ and $i \in \mathcal{V}$, is invertible.

Next, for $i \neq j$ and $i, j \in \mathcal{V}$, the cross covariance terms (18c) are written as

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

Let

$$\mathbf{\Pi}_{ij}(k+1) = \mathbf{\Pi}_{ij}(k) - \mathbf{\Gamma}_i\mathbf{\Gamma}_j^\top.$$

Then, the cross covariance update (18c) can be rewritten as

$$\mathbf{P}^{+}_{ij}(k+1) = \mathbf{\Phi}^i(k+1)\, \mathbf{\Pi}_{ij}(k+1)\, \mathbf{\Phi}^j(k+1)^\top. \tag{13}$$

Therefore, at time $k+2$, the propagated cross covariances terms for $i \neq j$ and $i, j \in \mathcal{V}$ are

$$\begin{aligned}
\mathbf{P}^{-}_{ij}(k+2) &= \mathbf{F}^i(k+1)\, \mathbf{P}^{+}_{ij}(k+1)\, \mathbf{F}^j(k+1)^\top \\
&= \mathbf{F}^i(k+1)\mathbf{\Phi}^i(k+1)\mathbf{\Pi}_{ij}(k+1)\mathbf{\Phi}^j(k+1)^\top\mathbf{F}^j(k+1)^\top \\
&= \mathbf{\Phi}^i(k+2)\, \mathbf{\Pi}_{ij}(k+1)\, \mathbf{\Phi}^j(k+2)^\top.
\end{aligned}$$

In short, the propagated and the updated cross covariance terms of the centralized EKF CL can be written as, respectively, (9) and (13) for all $k \in \{0, 1, \cdots\}$ where the $\mathbf{\Phi}^i(k)$ of each robot evolves according to (10) and

$$\mathbf{\Pi}_{ij}(k+1) = \begin{cases} \mathbf{\Pi}_{ij}(k), & \text{no relative measurement at } k+1, \\ \mathbf{\Pi}_{ij}(k) - \mathbf{\Gamma}_i\, \mathbf{\Gamma}_j^\top, & \text{otherwise,} \end{cases} \tag{14}$$

for $i,j\in\mathcal{V}$ and $i\neq j$.

Next, notice that the updated state estimate and covariance matrix in the new variables can be written as follows, for $i\in\mathcal{V}$,

$$\begin{aligned}\hat{\mathbf{x}}^{i+}(k+1)&=\hat{\mathbf{x}}^{i-}(k+1)+\mathbf{\Phi}^i(k+1)\,\mathbf{\Gamma}_i\,\bar{\mathbf{r}}^a,\\ \mathbf{P}^{i+}(k+1)&=\mathbf{P}^{i-}(k+1)-\mathbf{\Phi}^i(k+1)\mathbf{\Gamma}_i\,\mathbf{\Gamma}_i^\top\mathbf{\Phi}^i(k+1)^\top_,\end{aligned}\tag{15}$$

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

Using the alternative representations (9), (13), and (15) of the EKF CL, the decentralized implementation IMDCL is given in Algorithm 2. The IMDCL algorithm is developed by keeping a local copy of $\mathbf{\Pi}_{lj}$ at each agent $i\in\mathcal{V}$, that is, $\mathbf{\Pi}^i_{jl}$ for all $j\in\mathcal{V}\backslash\{N\}$ and $l\in\{j+1,\cdots,N\}$. Here, because of the symmetry of the covariance matrix only, for example, the upper triangular part of this matrix is needed to be stored and to be evolved. For example, for a group of $N=4$ robots, every agent maintains a copy of $\{\mathbf{\Pi}^i_{12}$, $\mathbf{\Pi}^i_{13}$, $\mathbf{\Pi}^i_{14}$, $\mathbf{\Pi}^i_{23}$, $\mathbf{\Pi}^i_{24}$, $\mathbf{\Pi}^i_{34}\}$. During the algorithm implementation, it is assumed that if $\mathbf{\Pi}^i_{jl}$ is not explicitly maintained by agent i, the agent substitutes the value of $(\mathbf{\Pi}^i_{lj})^\top$ for it.

In IMDCL, every agent $i\in\mathcal{V}$ initializes its own state estimate $\hat{\mathbf{x}}^{i+}(0)$, the error covariance matrix $\mathbf{P}^{i+}(0)$, $\mathbf{\Phi}^i(0)=\mathbf{I}_{n^i}$, and its local copies $\mathbf{\Pi}^i_{jl}(0)=\mathbf{0}_{n^j\times n^l}$, for $j\in\mathcal{V}\backslash\{N\}$ and $l\in\{j+1,\cdots,N\}$; see (20). At propagation stage, every agent evolves its local state estimate, error covariance and $\mathbf{\Phi}^i$, according to, respectively, (16a), (16b), (10); see (21). At every time step, when there is no exteroceptive measurement in the team, the local updated state estimates and error covariance matrices are replaced by their respective propagated counterparts, while each

$\mathbf{\Pi}^i_{jl}$, to respect (14), is kept unchanged; see (22). When there is a robot-to-robot measurement, examining (6), (5a), (11), (12b), and (12c) shows that agent a, the robot that made the relative measurement, can calculate these terms using its local $\mathbf{\Pi}^i_{jl}$ and acquiring $\hat{\mathbf{x}}^{b-}(k+1) \in \mathbb{R}^{n^b}$, $\mathbf{\Phi}^b(k+1) \in \mathbb{R}^{n^b \times n^b}$, and $\mathbf{P}^{b-}(k+1) \in \mathbb{M}_{n^b}$; see (23) and (24). Then, agent a can assume the role of the interim master and issue the update terms for other agents in the team; see (25). Using this update message and their local variables, then each agent $i \in \mathcal{V}$ can compute (12a) and obtain its local state updates of (15) and (14); see (27). Figure 6 demonstrates the information flow direction between agents while implementing the IMDCL algorithm.

The inclusion of absolute measurements in the IMDCL is straightforward. The agent making an absolute measurement is an interim master that can calculate the *update-message* using only its own data and then broadcast it to the team. Next, observe that the IMDCL algorithm is robust to permanent agent dropouts from the network. The operation only suffers from a processing cost until all agents become aware of the dropout. Also, notice that an external authority, for example, a search-and-rescue chief, who needs to obtain the location of any agent, can obtain this location update at any rate (s)he wishes to by communicating with that agent. This reduces the communication cost of the external authority.

The IMDCL algorithm works under the assumption that the message from the agent taking the relative measurement, the interim master, reaches the entire team. Any communication failure results in a mismatch between the local copies of $\mathbf{\Pi}_{lj}$ at the agents receiving and missing the communication message. Readers are referred to [34] for a variation of IMDCL that is robust to intermittent communication message dropouts. Such guarantees in [34] are provided by replacing the fully decentralized implementation with a partial decentralization where a shared memory stores and updates the $\mathbf{\Pi}_{lj}$, $j \in \mathcal{V}\backslash\{N\}$ and $l \in \{j+1, \cdots, N\}$.

Complexity analysis

For the sake of an objective performance evaluation, a study of the computational complexity, the memory usage, as well as communication cost per agent per time step of the IMDCL algorithm in terms of the size of the mobile-agent team N is provided next. At the propagation state of the IMDCL algorithm, the computations per agent are independent of the size of the team. However, at the update stage, for each measurement update, the computation of every agent is of order $N(N-1)/2$ due to (33c). Because multiple relative measurements are processed sequentially, the computational cost per agent at the completion of any update stage depends on the number of the relative measurements in the team, henceforth denoted by N_z. Then, the computational cost per agent is $O(N_z \times N^2)$, implying a computational complexity of order $O(N^4)$ for the worst case where all the agents take relative measurement with respect to all the other agents in the team, that is, $N_z = N(N-1)$. The memory cost per agent is of order $O(N^2)$, which, due to the recursive nature of the IMDCL algorithm, is independent of N_z. This cost is caused by the initialization (20) and update equation (33c), which are of order $N(N-1)/2$.

The analysis of the communication cost below is carried out for a multihop communication strategy. The IMDCL requires communication only in the update stage, where landmark robots should broadcast their landmark message to their respective master, and every agent should rebroadcast any update-message it receives. Let N_r be the number of the agents that have made a relative measurement at the current time, that is, $N_r \leq N$ is the number of current sequential interim masters. These robots should announce their identity and the number of their landmark robots to the entire team for sequential update cuing purpose, incurring a communication cost of order N_r per robot. Next, the team will proceed by sequentially processing the relative measurements. Every agent can be a landmark of $N_a \leq N_r$ agents and/or a master of $N_b \leq (N-1)$ agents. The updating procedure starts by a landmark robot sending its landmark message to its active interim master, resulting in a total communication cost of $O(N_a)$ per landmark robot at the end of update stage. Every active interim master passes an update message to the entire team, resulting in a total communication cost of $O(N_b)$ per robot. Because there are N_r masters, at the end of the update stage, every robot incurs a communication cost of $O(N_r \times N_b)$ to pass the update messages. Because $N_a, N_r < N_r \times N_b \leq N_z$, the total communication cost at the end of the update stage $O(N_z)$ per agent, implying a worst case broadcast cost of $O(N^2)$ per agent. If the communication range is unbounded, the broadcast cost per agent is $O(\max\{N_b, N_a\})$, with the worst case cost of $O(N)$. The communication message size of each agent in both single or multiple relative measurements is independent of the group size N. As such, for the worst case scenario the communication message size is of $O(1)$.

The results of the analysis above are summarized in Table I and are compared to those of a trivial decentralized implementation of the EKF for CL (denoted by T-DCL) in which every agent $i \in \mathcal{V}$ at the propagation stage computes (16)–using the broadcasted $\mathbf{F}^j(k)$ from every other team member $j \in \mathcal{V}\backslash\{i\}$–and at the update stage computes (19) and (18)–using the broadcast (a, b, $\mathbf{r}^a$, $\mathbf{S}_{ab}$, $\tilde{\mathbf{H}}_a$, $\tilde{\mathbf{H}}_b$, $\mathbf{R}^a$, $\mathbf{P}^{a\text{-}}$, $\mathbf{P}^{b\text{-}}$) from agent a that has made relative measurement from agent b. Agent

a calculates $\mathbf{S}_{ab}$, $\tilde{\mathbf{H}}_a$, $\tilde{\mathbf{H}}_b$ by requesting ($\hat{\mathbf{x}}^{b\text{-}}$, $\mathbf{P}^{b\text{-}}$) from agent b. Here, it is assume that multiple measurements are processed sequentially and that the communication procedure is multihop. Although the overall cost of the T-DCL algorithm is comparable with the IMDCL algorithm, this implementation has a more stringent communication connectivity condition, requiring a *strongly connected digraph* topology (that is, all the nodes on the communication graph can be reached by every other node on the graph) at each time step, regardless of whether there is a relative measurement incidence in the team. As an example, notice that the communication graph of Figure 2 is not strongly connected and as such the T-DCL algorithm can not be implemented whereas the IMDCL algorithm can be. Recall that the IMDCL algorithm needs no communication at the propagation stage and it only requires an existence of a spanning tree rooted at the agent making the relative measurement at the update stage. Finally, the IMDCL algorithm incurs less computational cost at the propagation stage.

Algorithm 3 presents an alternative IMDCL implementation where, instead of storing and evolving $\mathbf{\Pi}_{lj}$'s of the entire team, every agent only maintains the terms corresponding to its own cross covariances; see (28) and (29). For example, in a team of $N=4$, robot 1 maintains $\{\mathbf{\Pi}^1_{12}, \mathbf{\Pi}^1_{13}, \mathbf{\Pi}^1_{14}\}$ and robot 2 maintains $\{\mathbf{\Pi}^2_{21}, \mathbf{\Pi}^2_{23}, \mathbf{\Pi}^2_{24}\}$. However, now the interim master a needs to acquire the $\mathbf{\Pi}_{bj}$'s from the landmark robot b and calculate and broadcast $\mathbf{\Gamma}_i$, $i \in \mathcal{V}$ to the entire team; see (30), (31) and (33). In this alternative implementation, the processing and storage cost of every agent is reduced from $O(N^2)$ to $O(N)$, however the communication message size is increased from $O(1)$ to $O(N)$.

Tightly coupled versus loosely coupled DCL: a numerical comparison study

The IMDCL falls under the tightly coupled DCL classification. Figure 7 demonstrates the positioning accuracy (time history of the root-mean square error (RMSE) plot for 50 Monte-Carlo simulation runs) of this algorithm versus the loosely coupled EKF and covariance-intersection-based algorithm of [25] in the following scenario. for this numerical study, consider again the 3 mobile robots employed in the numerical example of "Cooperative localization via EKF" with motion as described in that section. The sensing scenario here is that, starting at $t = 10$ seconds, robot 3 takes persistent relative measurements alternating every 50 seconds from robot 1 to robot 2 and vice versa. As expected, the tightly coupled IMDCL algorithm produces more accurate position estimation results than those of the loosely coupled DCL algorithm of [25] (similar results can be observed for the heading estimation accuracy, which is omitted here for brevity).

In the algorithm of [25], every robot keeps an EKF estimation of its own pose. When a robot takes a relative pose measurement from another robot (here, also refer to the robot taking the relative measurement as the interim master), it acquires the current position estimate and the corresponding error covariance of the landmark robot to use along with its own current estimate and the current relative measurement to extract a new state estimate and the corresponding error covariance for the landmark robot. Next, the interim master robot transmits this new estimate to the landmark robot, which uses the covariance intersection method to fuse it, consistently, to its current pose estimate. It is interesting to notice that in this particular scenario, even though robot 3 has been taking all the relative measurements, it receives no benefit from such measurements, because only the landmark robots are updating their estimates; see Figure 7. Even though the positioning accuracy of algorithm [25] is lower, it only requires $O(1)$ computational cost per agent as compared to the $O(N^2)$ cost of the IMDCL algorithm. However, it also requires more complicated calculations to perform covariance intersection fusion. If the communication range of each agent covers the entire team, then interestingly the communication cost of these two algorithms is the same because both use an $O(1)$ landmark and update messages. However, if the communication range is bounded, the loosely coupled algorithm of [25] offers a more flexible and cost-effective communication policy.

Conclusions

This paper presented a brief review on cooperative localization as a strategy to increase the localization accuracy of a team of mobile agents with communication capabilities. This strategy relies on use of agent-to-agent relative measurements (no reliance on external features) as a feedback signal to *jointly estimate* the poses of the team members. In particular, this paper discussed challenges involved in designing decentralized cooperative localization algorithms. Moreover, it presented a decentralized cooperative localization algorithm that is exactly equivalent to the centralized EKF algorithm of [22]. In this decentralized algorithm, the propagation stage is fully decoupled, that is, the propagation is a local calculation and no intra-network communication is needed. The communication between agents is only required in the update stage when one agent makes a relative measurement with respect to another agent. The algorithm declares the agent made the measurement as interim master that can, by using the data acquired from the landmark agent, calculate, and deliver by broadcast the update terms for the rest of the team. Future extensions of this work includes handling message dropouts and asynchronous measurement updates.

References

[1] J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, June 1991.

[2] G. Dissanayake, P. Newman, H. F. Durrant-Whyte, , S. Clark, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[3] S. Cooper and H. Durrant-Whyte. A Kalman filter model for GPS navigation of land vehicles. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 157–163, Munich, Germany, 1994.

[4] J. S. Jennings, G. Whelan, and W. F. Evans. Cooperative search and rescue with a team of mobile robots. In *Int. Conf. on Advanced Robotics*, pages 193–200, 1997.

[5] A. Davids. Urban search and rescue robots: from tragedy to technology. *IEEE Intelligent Systems*, 17(2):81–83, March-April 2002.

[6] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, 2000.

[7] M. Trincavelli, M. Reggente, S. Coradeschi, A. Loutfi, H. Ishida, and A. J. Lilienthal. Towards environmental monitoring with mobile robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 2210–2215, 2008.

[8] A. Bahr, M. R. Walter, and J. J. Leonard. Consistent cooperative localization. In *IEEE Int.*

Conf. on Robotics and Automation, pages 8908–8913, Kobe, Japan, May 2009.

[9] H. Mokhtarzadeh and D Gebre-Egziabher. Cooperative inertial navigation. *Navigation: Journal of the Institute of Navigation*, 61(2):77–94, 2014.

[10] R. Kurazume, S. Nagata, and S. Hirose. Cooperative positioning with multiple robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 1250–1257, San Diego, CA, May 8–13 1994.

[11] I. Rekleitis, G. Dudek, and E. Milios. Multi-robot collaboration for robust exploration. In *IEEE Int. Conf. on Robotics and Automation*, pages 3164–3169, 2000.

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

[13] A. Howard, M. Matark, and G. Sukhatme. Localization for mobile robot teams using maximum likelihood estimation. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, volume 1, pages 434–439, 2002.

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

[15] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344, 2000.

[16] A. Howard, M. J. Mataric, and G. S. Sukhatm. Putting the 'I' in 'team': An ego-centric approach to cooperative localization. In *IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 868–874, 2003.

[17] A. Prorok and A. Martinoli. A reciprocal sampling algorithm for lightweight distributed

multi-robot localization. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 3241–3247, 2011.

[18] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky. Nonparametric belief propagation for self-localization of sensor networks. *IEEE Journal of Selected Areas in Communications*, 23(4):809–819, 2005.

[19] J. Nilsson, D. Zachariah, I. Skog, and P. Händel. Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging. *EURASIP Journal on Advances in Signal Processing*, 2013(164), 2013.

[20] 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*, pages 299–308, Fukuoka, Japan, June 2002.

[21] 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*, pages 3761–3765, Chicago, IL, September 2014.

[22] S. I. Roumeliotis and G. A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.

[23] C. T. Leondes, editor. *Advances in Control Systems Theory and Application*, volume 3. Academic Press, New York, 1966.

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

[25] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis. Decentralized

multi-robot cooperative localization using covariance intersection. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 1412–1417, Tokyo, Japan, 2013.

[26] P. O. Arambel, C. Rago, and R. K. Mehra. Covariance intersection algorithm for distributed spacecraft state estimation. In *American Control Conference*, pages 4398–4403, Arlington, VA, 2001.

[27] H. Li and F. Nashashibi. Cooperative multi-vehicle localization using split covariance intersection filter. *IEEE Intelligent Transportation Systems Magazine*, 5(2):33–44, 2013.

[28] D. Marinescu, N. O'Hara, and V. Cahill. Data incest in cooperative localisation with the common past-invariant ensemble kalman filter. pages 68–76, Istanbul, Turkey, 2013.

[29] S. J. Julier and J. K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *American Control Conference*, pages 2369–2373, Albuquerque, NM, 1997.

[30] S. J. Julier and J. K. Uhlmann. Simultaneous localisation and map building using split covariance intersection. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 1257–1262, Maui, HI, 2001.

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

[32] K. Y. K. Leung, T. D. Barfoot, and H. H. T. Liu. Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach. *IEEE Transactions on Robotics*, 26(1):62–77, 2010.

[33] L. Paull, M. Seto, and J. J. Leonard. Decentralized cooperative trajectory estimation for

autonomous underwater vehicles. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 184–191, 2014.

[34] S. S. Kia, S. Rounds, and S. Mart/'/inez. Cooperative Localization under message dropouts via a partially decentralized EKF scheme. In *IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, May 2015.

[35] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: Part i. 13(2):99–110, 2006.

Algorithm 1 EKF CL (centralized)

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

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

Iteration k

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

$$\hat{\mathbf{x}}^{i-}(k+1)=\mathbf{f}^i(\hat{\mathbf{x}}^{i+}(k),\mathbf{u}^i(k)), \tag{16a}$$

$$\mathbf{P}^{i-}(k+1)=\mathbf{F}^i(k)\mathbf{P}^{i+}(k)\mathbf{F}^i(k)^\top+\mathbf{G}^i(k)\mathbf{Q}^i(k)\mathbf{G}^i(k)^\top, \tag{16b}$$

$$\mathbf{P}^{-}_{ij}(k+1)=\mathbf{F}^i(k)\mathbf{P}^{+}_{ij}(k)\mathbf{F}^j(k)^\top,\quad j\in\mathcal{V}\backslash\{i\}. \tag{16c}$$

2: Update: While there are no relative measurements no update happens, that is,

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

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

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

and

$$\begin{aligned}\mathbf{S}_{ab}=\,&\mathbf{R}^a(k+1)+\tilde{\mathbf{H}}_a(k+1)\mathbf{P}^{a-}(k+1)\tilde{\mathbf{H}}_a(k+1)^\top+\tilde{\mathbf{H}}_b(k+1)\mathbf{P}^{b-}(k+1)\tilde{\mathbf{H}}_b(k+1)^\top\\&-\tilde{\mathbf{H}}_b(k+1)\mathbf{P}^{-}_{ba}(k+1)\tilde{\mathbf{H}}_a(k+1)^\top-\tilde{\mathbf{H}}_a(k+1)\mathbf{P}^{-}_{ab}(k+1)\tilde{\mathbf{H}}_b(k+1)^\top.\end{aligned} \tag{17}$$

The estimation updates for the centralized EKF are

$$\hat{\mathbf{x}}^{i+}(k+1)=\hat{\mathbf{x}}^{i-}(k+1)+\mathbf{K}_i(k+1)\mathbf{r}^a(k+1), \tag{18a}$$

$$\mathbf{P}^{i+}(k+1)=\mathbf{P}^{i-}(k+1)-\mathbf{K}_i(k+1)\mathbf{S}_{ab}(k+1)\mathbf{K}_i(k+1)^\top, \tag{18b}$$

$$\mathbf{P}^{+}_{ij}(k+1)=\mathbf{P}^{-}_{ij}(k+1)-\mathbf{K}_i(k+1)\mathbf{S}_{ab}(k+1)\mathbf{K}_j(k+1)^\top, \tag{18c}$$

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

$$\mathbf{K}_i=(\mathbf{P}^{-}_{ib}(k+1)\tilde{\mathbf{H}}_b^\top-\mathbf{P}^{-}_{ia}(k+1)\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}{}^{-1}. \tag{19}$$

3: $k\leftarrow k+1$

Algorithm 2 IMDCL

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

$$\hat{\mathbf{x}}^{i+}(0) \in \mathbb{R}^{n^i}, \mathbf{P}^{i+}(0) \in \mathbb{M}_{n^i}, \quad \mathbf{\Phi}^i(0) = \mathbf{I}_{n^i}, \quad \mathbf{\Pi}^i_{jl}(0) = \mathbf{0}_{n^l \times n^j}, \; j \in \mathcal{V}\backslash\{N\}, \; l \in \{j+1, \cdots, N\}. \tag{20}$$

Iteration k

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

$$\hat{\mathbf{x}}^{i-}(k+1) = \mathbf{f}^i(\hat{\mathbf{x}}^{i+}(k), \mathbf{u}^i(k)), \quad \mathbf{P}^{i-}(k+1) = \mathbf{F}^i(k)\mathbf{P}^{i+}(k)\mathbf{F}^i(k)^\top + \mathbf{G}^i(k)\mathbf{Q}^i(k)\mathbf{G}^i(k)^\top, \quad \mathbf{\Phi}^i(k+1) = \mathbf{F}^i(k)\mathbf{\Phi}^i(k). \tag{21}$$

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

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i-}(k+1), \quad \mathbf{P}^{i+}(k+1) = \mathbf{P}^{i-}(k+1), \quad \mathbf{\Pi}^i_{jl}(k+1) = \mathbf{\Pi}^i_{lj}(k), \quad j \in \mathcal{V}\backslash\{N\}, \; l \in \{j+1, \cdots, N\}. \tag{22}$$

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

$$\textit{landmark-message} = \left(\hat{\mathbf{x}}^{b-}(k+1), \mathbf{\Phi}^b(k+1), \mathbf{P}^{b-}(k+1)\right). \tag{23}$$

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

$$\mathbf{r}^a = \mathbf{z}_{ab} - \mathbf{h}_{ab}(\hat{\mathbf{x}}^{a-}, \hat{\mathbf{x}}^{b-}), \tag{24a}$$

$$\mathbf{S}_{ab} = \mathbf{R}^a + \tilde{\mathbf{H}}_a \mathbf{P}^{a-} \tilde{\mathbf{H}}_a^\top + \tilde{\mathbf{H}}_b^\top \mathbf{P}^{b-} \tilde{\mathbf{H}}_b - \tilde{\mathbf{H}}_a \mathbf{\Phi}^a \mathbf{\Pi}^a_{ab} \mathbf{\Phi}^{b\top} \tilde{\mathbf{H}}_b^\top - \tilde{\mathbf{H}}_b \mathbf{\Phi}^b \mathbf{\Pi}^a_{ba} \mathbf{\Phi}^{a\top} \tilde{\mathbf{H}}_a^\top, \tag{24b}$$

$$\mathbf{\Gamma}_a = ((\mathbf{\Phi}^a)^{-1} \mathbf{\Phi}^a \mathbf{\Pi}^a_{ab} \mathbf{\Phi}^{b\top} \tilde{\mathbf{H}}_b^\top - (\mathbf{\Phi}^a)^{-1} \mathbf{P}^{a-} \tilde{\mathbf{H}}_a^\top) \mathbf{S}_{ab}^{-\frac{1}{2}}, \quad \mathbf{\Gamma}_b = ((\mathbf{\Phi}^b)^{-1} \mathbf{P}^{b-} \tilde{\mathbf{H}}_b^\top - \mathbf{\Pi}^a_{ba} \mathbf{\Phi}^{a\top} \tilde{\mathbf{H}}_a^\top) \mathbf{S}_{ab}^{-\frac{1}{2}}, \tag{24c}$$

where $\tilde{\mathbf{H}}_a(k+1) = \tilde{\mathbf{H}}_a(\hat{\mathbf{x}}^{a-}, \hat{\mathbf{x}}^{b-})$ and $\tilde{\mathbf{H}}_b(k+1) = \tilde{\mathbf{H}}_b(\hat{\mathbf{x}}^{a-}, \hat{\mathbf{x}}^{b-})$ are obtained using (6).

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

$$\textit{update-message} = \left(a, b, \bar{\mathbf{r}}^a, \mathbf{\Gamma}_a, \mathbf{\Gamma}_b, \mathbf{\Phi}^{b\top} \tilde{\mathbf{H}}_b^\top \mathbf{S}_{ab}^{-\frac{1}{2}}, \mathbf{\Phi}^{a\top} \tilde{\mathbf{H}}_a^\top \mathbf{S}_{ab}^{-\frac{1}{2}}\right). \tag{25}$$

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

$$\mathbf{\Gamma}_j = \mathbf{\Pi}^i_{jb} \mathbf{\Phi}^{b\top} \tilde{\mathbf{H}}_b^\top \mathbf{S}_{ab}^{-\frac{1}{2}} - \mathbf{\Pi}^i_{ja} \mathbf{\Phi}^{a\top} \tilde{\mathbf{H}}_a^\top \mathbf{S}_{ab}^{-\frac{1}{2}}, \tag{26}$$

and then updates the following variables

$$\hat{\mathbf{x}}^{i+}(k+1) = \hat{\mathbf{x}}^{i-}(k+1) + \mathbf{\Phi}^i(k+1)\, \mathbf{\Gamma}_i\, \bar{\mathbf{r}}^a, \tag{27a}$$

$$\mathbf{P}^{i+}(k+1) = \mathbf{P}^{i-}(k+1) - \mathbf{\Phi}^i(k+1) \mathbf{\Gamma}_i \mathbf{\Gamma}_i^\top \mathbf{\Phi}^i(k+1)^\top, \tag{27b}$$

$$\mathbf{\Pi}^i_{jl}(k+1) = \mathbf{\Pi}^i_{jl}(k) - \mathbf{\Gamma}_j \mathbf{\Gamma}_l^\top, \quad j \in \mathcal{V}\backslash\{N\}, l \in \{j+1, \cdots, N\}. \tag{27c}$$

3: $k \leftarrow k+1$

Algorithm 3 Alternative IMDCL (larger communication message size in favor of lower computation and storage cost per agent)

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

$$\hat{\mathbf{x}}^{i+}(0)\in\mathbb{R}^{n^i},\mathbf{P}^{i+}(0)\in\mathbb{M}_{n^i},\quad \mathbf{\Phi}^i(0)=\mathbf{I}_{n^i},\quad \mathbf{\Pi}^i_{ij}(0)=\mathbf{0}_{n^i\times n^j},\; j\in\mathcal{V}\backslash\{i\}. \tag{28}$$

Iteration k

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

$$\hat{\mathbf{x}}^{i\text{-}}(k+1)=\mathbf{f}^i(\hat{\mathbf{x}}^{i+}(k),\mathbf{u}^i(k)),\quad \mathbf{P}^{i\text{-}}(k+1)=\mathbf{F}^i(k)\mathbf{P}^{i+}(k)\mathbf{F}^i(k)^\top+\mathbf{G}^i(k)\mathbf{Q}^i(k)\mathbf{G}^i(k)^\top,\quad \mathbf{\Phi}^i(k+1)=\mathbf{F}^i(k)\mathbf{\Phi}^i(k). \tag{29}$$

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

$$\hat{\mathbf{x}}^{i+}(k+1)=\hat{\mathbf{x}}^{i\text{-}}(k+1),\quad \mathbf{P}^{i+}(k+1)=\mathbf{P}^{i\text{-}}(k+1),\quad \mathbf{\Pi}^i_{ij}(k+1)=\mathbf{\Pi}^i_{ij}(k),\quad j\in\mathcal{V}\backslash\{i\}.$$

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

$$\textit{landmark-message}=\Big(\hat{\mathbf{x}}^{b\text{-}}(k+1),\mathbf{\Phi}^b(k+1),\mathbf{P}^{b\text{-}}(k+1),\mathbf{\Pi}^b_{bj}(k)\text{ where } j\in\mathcal{V}\backslash\{a,b\}\Big). \tag{30}$$

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

$$\mathbf{r}^a=\mathbf{z}_{ab}-\mathbf{h}_{ab}(\hat{\mathbf{x}}^{a\text{-}},\hat{\mathbf{x}}^{b\text{-}}), \tag{31a}$$

$$\mathbf{S}_{ab}=\mathbf{R}^a+\tilde{\mathbf{H}}_a\mathbf{P}^{a\text{-}}\tilde{\mathbf{H}}_a^\top+\tilde{\mathbf{H}}_b^\top\mathbf{P}^{b\text{-}}\tilde{\mathbf{H}}_b-\tilde{\mathbf{H}}_a\mathbf{\Phi}^a\mathbf{\Pi}^a_{ab}\mathbf{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top-\tilde{\mathbf{H}}_b\mathbf{\Phi}^b(\mathbf{\Pi}^a_{ab})^\top\mathbf{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top, \tag{31b}$$

$$\mathbf{\Gamma}_a=((\mathbf{\Phi}^a)^{-1}\mathbf{\Phi}^a\mathbf{\Pi}^a_{ab}\mathbf{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top-(\mathbf{\Phi}^a)^{-1}\mathbf{P}^{a\text{-}}\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}^{-\frac{1}{2}},\quad \mathbf{\Gamma}_b=((\mathbf{\Phi}^b)^{-1}\mathbf{P}^{b\text{-}}\tilde{\mathbf{H}}_b^\top-(\mathbf{\Pi}^a_{ab})^\top\mathbf{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top)\mathbf{S}_{ab}^{-\frac{1}{2}}, \tag{31c}$$

$$\mathbf{\Gamma}_j=(\mathbf{\Pi}^b_{bj})^\top\mathbf{\Phi}^{b\top}\tilde{\mathbf{H}}_b^\top\mathbf{S}_{ab}^{-\frac{1}{2}}-(\mathbf{\Pi}^i_{aj})^\top\mathbf{\Phi}^{a\top}\tilde{\mathbf{H}}_a^\top\mathbf{S}_{ab}^{-\frac{1}{2}},\quad j\in\mathcal{V}\backslash\{a,b\}, \tag{31d}$$

where $\tilde{\mathbf{H}}_a(k+1)=\tilde{\mathbf{H}}_a(\hat{\mathbf{x}}^{a\text{-}},\hat{\mathbf{x}}^{b\text{-}})$ and $\tilde{\mathbf{H}}_b(k+1)=\tilde{\mathbf{H}}_b(\hat{\mathbf{x}}^{a\text{-}},\hat{\mathbf{x}}^{b\text{-}})$ are obtained using (6).

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

$$\textit{update-message}=\Big(a,b,\bar{\mathbf{r}}^a,\mathbf{\Gamma}_1,\cdots,\mathbf{\Gamma}_N\Big). \tag{32}$$

Every agent $i\in\mathcal{V}$, upon receiving the *update-message*, updates the following variables

$$\hat{\mathbf{x}}^{i+}(k+1)=\hat{\mathbf{x}}^{i\text{-}}(k+1)+\mathbf{\Phi}^i(k+1)\,\mathbf{\Gamma}_i\,\bar{\mathbf{r}}^a, \tag{33a}$$

$$\mathbf{P}^{i+}(k+1)=\mathbf{P}^{i\text{-}}(k+1)-\mathbf{\Phi}^i(k+1)\mathbf{\Gamma}_i\mathbf{\Gamma}_i^\top\mathbf{\Phi}^i(k+1)^\top, \tag{33b}$$

$$\mathbf{\Pi}^i_{ij}(k+1)=\mathbf{\Pi}^i_{ij}(k)-\mathbf{\Gamma}_i\mathbf{\Gamma}_j^\top,\quad j\in\mathcal{V}\backslash\{i\}. \tag{33c}$$

3: $k\leftarrow k+1$

TABLE I: Complexity analysis per agent of the IMDCL algorithm (denoted by IM-DCL) compared to that of the trivial decentralized implementation of EKF for CL (denoted by T-DCL) introduced in Subsection .

	Computation		Storage		Broadcast*		Message Size		Connectivity	
Algorithm	IM-DCL	T-DCL	IM-DCL	T-DCL	IM-DCL	T-DCL	IM-DCL	T-DCL	IM-DCL	T-DCL
Propagation	$O(1)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	0	$O(N)$	0	$O(1)$	None	strongly connected digraph
Update per N_z relative measur.	$O(N_z \times N^2)$	$O(N_z \times N^2)$	$O(N^2)$	$O(N^2)$	$O(N_z)$	$O(N_z)$	$O(1)$	$O(1)$	interim master can reach all the agents	
Overall worst case	$O(N^4)$	$O(N^4)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(1)$	$O(1)$		

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

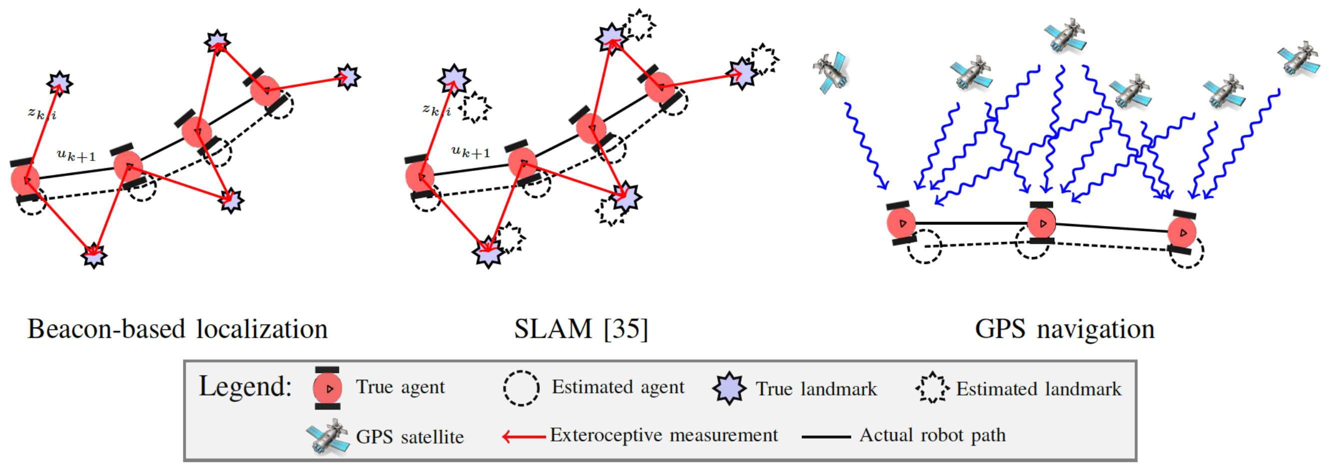
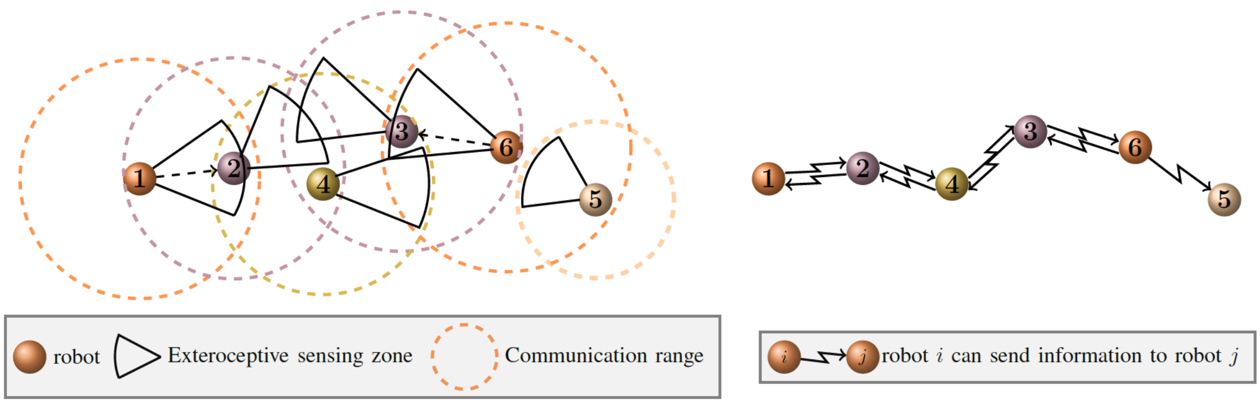


Figure 1: Schematic representation of common probabilistic localization techniques for mobile platforms: In beacon-based localization, the map of the area is known and there are pre-installed beacons or landmarks with known locations and identities. By taking relative measurements with respect to these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



(a) Communication and exteroceptive sensing ranges

(b) Communication graph

Figure 2: A multihop communication scenario for the multi-robot team. (a) shows the communication and measurement ranges. Here, robots 1 and 6 make relative measurements, respectively, of robots 2 and 3. (b) shows the communication graph generated using the communication ranges given in plot (a). Here the robot at the head of an arrow can send information to the robot at the tip of the arrow. As this graph shows, each of robots 1 and 6 can pass communication message to the entire team via a multihop strategy.

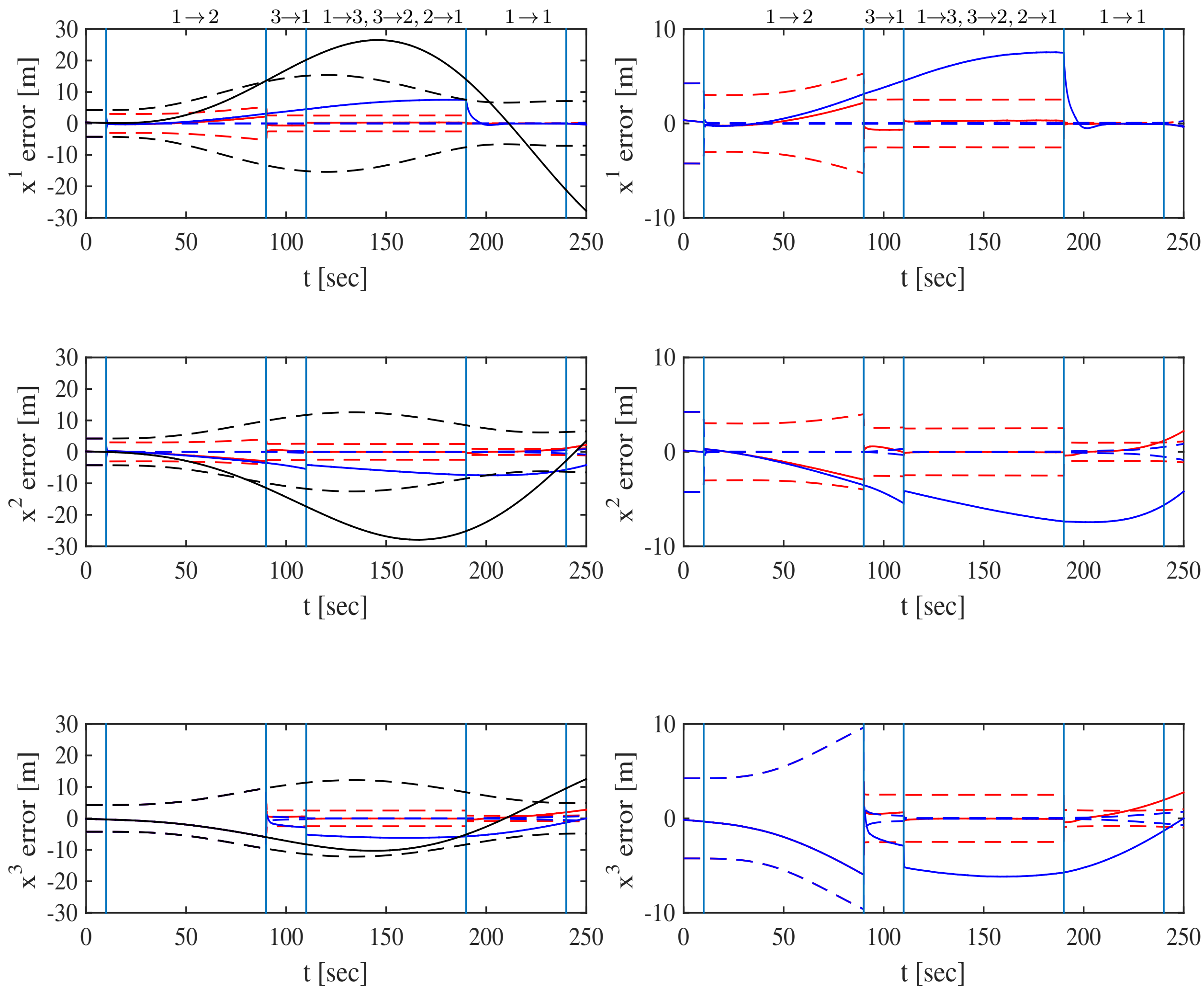
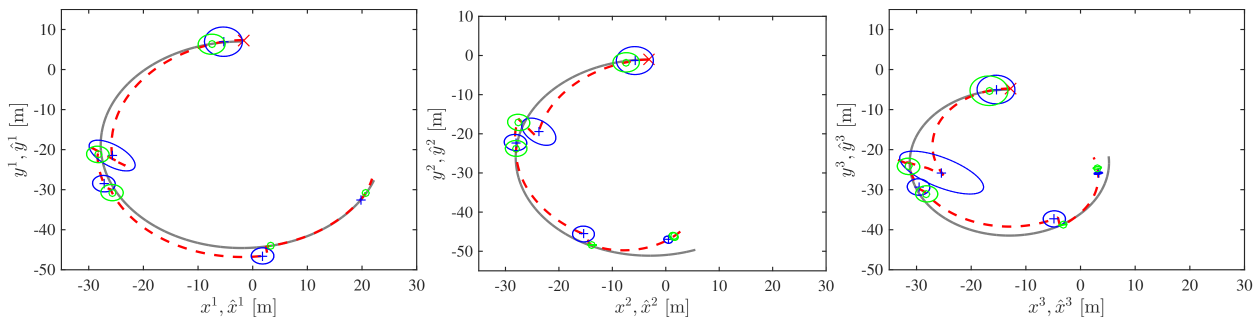


Figure 3: Estimation error (solid line) and 3σ error bounds (dashed lines) in the $x-$coordinate variable for 3 robots moving on a flat terrain when they (a) only propagate their equations of motion using self-motion measurements (black plots), (b) employ cooperative localization ignoring past correlations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



Robot 1 Robot 2 Robot3

Figure 4: Trajectories of the robots for the simulation study of Figure 3. Here, the gray curve is the ground truth. The red curve is the estimates of the trajectory by implementing a EKF CL. The blue (resp. green) ellipses show the 95% uncertainty regions for the estimations at 2 seconds before (resp. after) any change in the measurement scenario (see Figure 3)

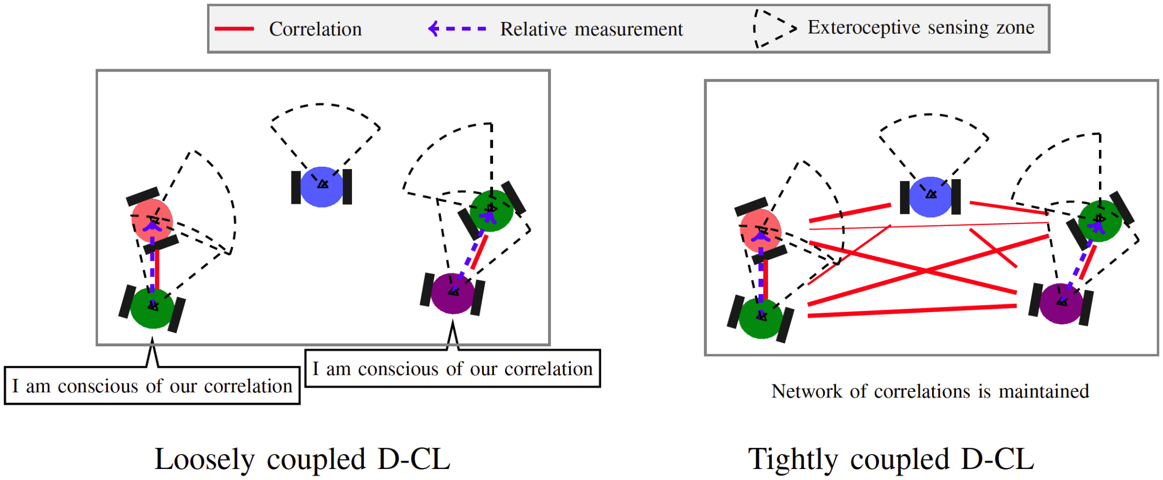
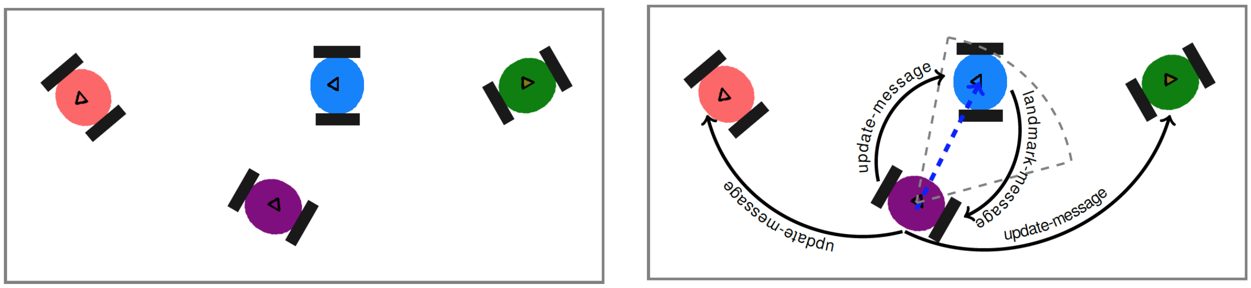


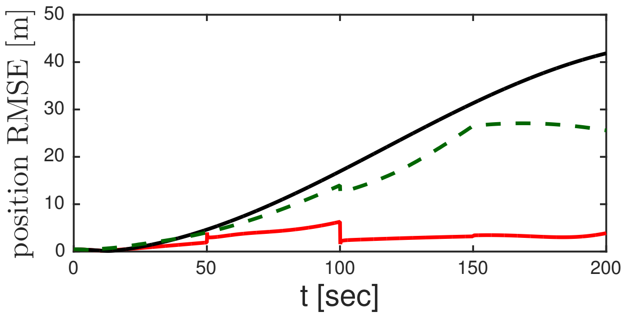
Figure 5: Schematic representation of the DCL classification based on the type of accounting for past correlations.



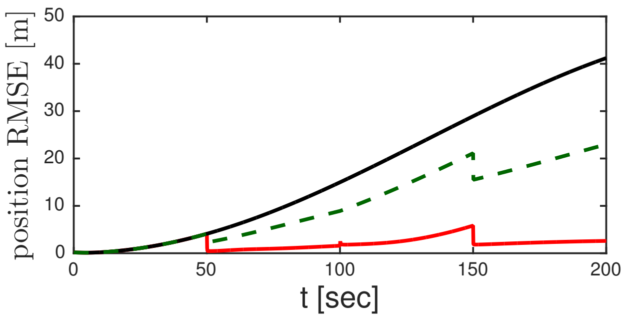
Propagation stage for all $k \in \{0, 1, 2, \cdots\}$

Update stage

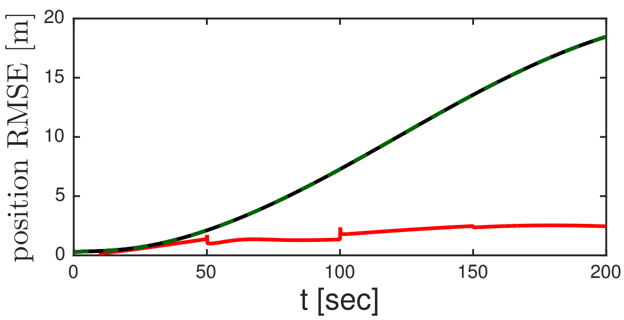
Figure 6: The in-network information flow of the IMDCL algorithm. In the IMDCL algorithm, communication is only needed in the update stage when the team members use a robot-to-robot relative measurement feedback to correct their pose estimation. Here, it is assumed that all the team members are in the communication range of the interim master robot.



Robot 1



Robot 2



Robot3

Figure 7: A comparison study between the positioning accuracy of three robots employing the interim master decentralized cooperative localization (IMDCL) algorithm (red plots), with that from the extended Kalman filter covariance intersection based CL algorithm of [25] (dashed green plot). The curves in black show the positioning accuracy when the robots do not use any cooperative localization. As expected, the IMDCL algorithm by keeping an accurate account of the cross covariances produces more accurate localization results than the algorithm of [25] . However, this higher accuracy comes with higher communication and processing cost per robot. Notice here that using algorithm of [25] robot 3 does not get to update its estimation equations.

Sidebar 1

Further Reading

A performance analysis of an EKF CL for a team of homogeneous robots moving on a flat terrain, with the same level of uncertainty in their proprioceptive measurements and exteroceptive sensors that measure relative pose, is provided in [S1] and [S2]. Interestingly, [S1] shows that the rate of uncertainty growth decreases as the size of the robot team increases, but is subject to the law of diminishing returns. Moreover, [S2] shows that the upper bound on the rate of uncertainty growth is independent of the accuracy or the frequency of the robot-to-robot measurements. The consistency of EKF CL from the perspective of observability is studies in [S3]. [S3] analytically shows that the error-state system model employed in the standard EKF CL always has an observable subspace of higher dimension than that of the actual nonlinear CL system. This results in an unjustified reduction of the EKF covariance estimates in directions of the state space where no information is available, and thus leads to inconsistency. To address this problem, [S3] adopts an observability-based methodology for designing consistent estimators in which the linearization points are selected to ensure a linearized system model with an observable subspace of the correct dimension. More results on observability analysis of CL can be found in [22], [S4], [S5]. The use of an observability analysis to explicitly design an active local path-planning algorithm for unmanned aerial vehicles implementing a bearing-only CL is discussed in [S6]. The necessity for an initialization procedure for CL is discussed in [S7], and it is shown that, because of system nonlinearities and the periodicity of the orientation, initialization errors can lead to erroneous results in covariance-based filters. An initialization procedure for the state estimate in a CL scenario based on ranging and dead reckoning is studied in [S8].

References

[S1] S. I. Roumeliotis and A. I. Mourikis. Propagation of uncertainty in cooperative multirobot localization: Analysis and experimental results. *Autonomous Robots*, 17(1):1573–7527, 2004.

[S2] A. I. Mourikis and S. I. Roumeliotis. Performance analysis of multirobot cooperative localization. *IEEE Transactions on Robotics*, 22(4):666–681, 2006.

[S3] G. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis. Observability-based consistent EKF estimators for multi-robot cooperative localization. *Autonomous Robots*, 30(1):37–58, 2011.

[S4] A. Martinelli and R. Siegwart. Observability analysis for mobile robot localization. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 1471–1476, 2005.

[S5] R. Sharma, R. W. Beard, C. N. Taylor, and S. Quebe. Graph-based observability analysis of bearing-only cooperative localization. *IEEE Transactions on Robotics*, 28(2):522–529, 2012.

[S6] R. Sharma. *Bearing-Only Cooperative-Localization and Path-Planning of Ground and Aerial Robots*. PhD thesis, Brigham Young University, 2011.

[S7] N. Trawny. *Cooperative localization: On motion-induced initialization and joint state estimation under communication constraint*. PhD thesis, University of Minnesota, 2010.

[S8] J. Nilsson and P. Händel. Recursive bayesian initialization of localization based on ranging and dead reckoning. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, pages 1399–1404, 2013.

Authors Information

Solmaz S. Kia is an Assistant Professor in the Department of Mechanical and Aerospace Engineering, University of California, Irvine (UCI). She obtained her Ph.D. degree in Mechanical and Aerospace Engineering from UCI, in 2009, and her M.Sc. and B.Sc. in Aerospace Engineering from the Sharif University of Technology, Iran, in 2004 and 2001, respectively. She was a senior research engineer at SySense Inc., El Segundo, CA from Jun. 2009-Sep. 2010. She held postdoctoral positions in the Department of Mechanical and Aerospace Engineering at the UC San Diego and UCI. Her main research interests, in a broad sense, include distributed optimization/coordination/estimation, nonlinear control theory, and probabilistic robotics.

Stephen Rounds is a Research Engineer with NavCom Technology, a John Deere company. He is responsible for identifying and supporting new navigation technologies for the company and developing and maintaining the intellectual property portfolio of the company. Prior to working with John Deere, he worked with multiple defense contractors, with a special emphasis on GPS-denied navigation, GPS antijamming protection, and other sensor fusion applications. He holds a B.S. degree in physics from Stevens Institute of Technology in Hoboken, N.J., and an M.S. degree in nuclear physics from Yale University. He is the Chairman of the Southern California section of the Institute Of Navigation, holds multiple patents in the navigation field, and has numerous publications in various aerospace and navigation forums.

Sonia Martínez is a Professor with the department of Mechanical and Aerospace Engineering at the University of California, San Diego. She received the Ph.D. degree in Engineering

Mathematics from the Universidad Carlos III de Madrid, Spain, in May 2002. Following a year as a Visiting Assistant Professor of Applied Mathematics at the Technical University of Catalonia, Spain, she obtained a Postdoctoral Fulbright Fellowship and held appointments at the Coordinated Science Laboratory of the University of Illinois, Urbana-Champaign during 2004, and at the Center for Control, Dynamical systems and Computation (CCDC) of the University of California, Santa Barbara during 2005. In a broad sense, her main research interests include the control of networked systems, multi-agent systems, nonlinear control theory, and robotics. For her work on the control of underactuated mechanical systems she received the Best Student Paper award at the 2002 IEEE Conference on Decision and Control. She was the recipient of a NSF CAREER Award in 2007. For the paper "Motion coordination with Distributed Information," co-authored with Jorge Cortés and Francesco Bullo, she received the 2008 Control Systems Magazine Outstanding Paper Award. She has served on the editorial boards of the European Journal of Control (2011-2013), and currently serves on the editorial board of the Journal of Geometric Mechanics and IEEE Transactions on Control of Networked Systems.