2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Decentralized Multi-robot Cooperative Localization using Covariance Intersection

Luis C. Carrillo-Arce[†], Esha D. Nerurkar[‡], José L. Gordillo[†], and Stergios I. Roumeliotis[‡]

Abstract— **In this paper, we present a Covariance Intersection (CI)-based algorithm for reducing the processing and communication complexity of multi-robot Cooperative Localization (CL). Specifically, for a team of N robots, our proposed approximate CI-based CL approach has processing and communication complexity only linear, $O(N)$, in the number of robots. Moreover, and in contrast to alternative approximate methods, our approach is provably consistent, can handle asynchronous communication, and does not place any restriction on the robots' motion. We test the performance of our proposed approach in both simulations and experimentally, and show that it outperforms the existing linear-complexity split CI-based CL method.**

I. Introduction

Accurate multi-robot localization, i.e., precisely determining the robots' poses (position and orientation), is pivotal for the successful execution of higher-level tasks such as target tracking, path planning, mapping, etc. In GPS-denied environments (e.g., outer-space, indoors, urban canyons, and underwater), a commonly-used localization technique is Cooperative Localization. In CL, groups of communicating robots use *all* their individual motion measurements (e.g., velocity) and robot-to-robot relative measurements (e.g., distance, bearing, and orientation) to *jointly* estimate their poses, hence improving the localization accuracy for the entire team [16].

However, the improved estimation accuracy in CL is achieved at the cost of increased computational and communication requirements. For example, consider the Extended Kalman filter (EKF)-based CL [16], where a team of N robots obtains a maximum of $O(N^2)$ relative measurements[1] at each time step. Here, the computational complexity for processing a single measurement is $O(N^2)$, resulting in an overall cost of $O(N^4)$ per time step. Thus, as the size of the robot team increases, these high processing requirements can prohibit real-time operation. Moreover, since each robot has to communicate its measurement information to the team or a fusion center, the communication cost of CL can be as high as $O(N)$ per robot, at each time step. In communication-bandwidth-constrained applications, these requirements may far exceed the power/battery resources of the robot team. Additionally, for robot teams navigating in large environments, the robots may not be able to exchange information with the fusion center, or they may be able to communicate with only a part of the team at each time step (asynchronous communication).

To address these issues, various decentralized estimation algorithms have been developed in the literature. In particular, some of the proposed approaches are able to distribute the processing for only a part of the CL algorithm [16], [2], while other fully-distributed algorithms require synchronous communication among the robots [14]. Approaches that can handle asynchronous communication have been proposed [11], [15], [10], but these are based on approximations (ignoring cross-correlations between robot pose estimates) which may lead to inconsistent estimation.

In this paper, we present an approximate decentralized algorithm for CL, with reduced processing and communication requirements. In our proposed approach, instead of maintaining the joint state of all robots along with the $O(N^2)$ covariance matrix (as is done in the standard EKF-based CL), each robot i locally maintains an estimate for only its *own* state and the corresponding covariance. When robot i obtains a relative measurement to another robot j, robot i uses this measurement and its local information to generate a state-covariance estimate pair for robot j. This information is then communicated to robot j. In order to ensure consistent information fusion, robot j uses the CI algorithm [7] to fuse the received estimate (from robot i) with its local information, thus obtaining an improved estimate for itself. The key contributions of our approach are as follows:

- Since each robot maintains only its own state estimate and covariance, the processing and communication requirements of our proposed algorithm are only $O(N)$ per robot, at each time step.
- Our approach can efficiently handle *asynchronous* communication since it places no restrictions on the team's underlying communication network or the robots' motion profile. Robots i and j need to communicate only when they obtain relative measurements to each other.
- The proposed estimator is *provably consistent*[2]. Even though the robots do not maintain a joint state, each

* This work was supported by Consejo Nacional de Ciencia y Tecnología (CONACyT), the Center for Robotics and Intelligent Systems (CRIS), the e-Robots Research Chair from Tecnológico de Monterrey, the University of Minnesota through the Digital Technology Center (DTC), and the National Science Foundation (IIS-0643680). E. D. Nerurkar was supported by the UMN Doctoral Dissertation Fellowship

[†]L. C. Carrillo-Arce and J. L. Gordillo are with the Center for Robotics and Intelligent Systems (CRIS), Tecnológico de Monterrey, Monterrey, NL 64849, MEX {lc.carrillo.phd.mty|jlgordillo}[at]itesm.mx

[‡]E. D. Nerurkar and S. I. Roumeliotis are with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA {nerurkar|stergios}[at]cs.umn.edu

[1] This corresponds to the scenario where each robot obtains a relative measurement to all robots in the team.

[2] A state estimator is consistent if the estimation errors are zero-mean and have covariance equal to the one calculated by the filter [3].

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

robot is able to generate consistent estimates by using the CI algorithm.

II. Related Work

Existing work on CL can be classified based on: (i) the estimation architecture used (centralized [5] vs. decentralized [14]), and (ii) the estimation algorithm used (e.g., Maximum Likelihood (ML) [5], Maximum A Posteriori (MAP) [14], Particle Filter [4], CI [1], Split CI (SCI) [10], and EKF [16]). In what follows, we focus on both optimal (i.e., methods that utilize all available measurement information) and approximate decentralized CL approaches, developed for reducing the computational and/or communication requirements of CL.

Optimal EKF-based and information filter-based CL algorithms, that distribute the processing of proprioceptive measurements, have been proposed by Roumeliotis and Bekey [16], and Bailey *et al.* [2], respectively. However, these approaches are able to distribute only a part of the estimation algorithm since the processing of exteroceptive measurements for both approaches is centralized. A fully-distributed MAP-based approach is presented by Nerurkar *et al.* in [14], but it requires synchronous communication among robots, i.e., each robot has to communicate with the team at every time step.

Approximate CL approaches include the measurement-selection technique of Mourikis and Roumeliotis [13] where robots select and process only a subset of the available measurements, subject to communication and processing constraints. In [9], each robot maintains the joint-state but processes only its own locally-available measurements. Periodically, robots within communication range fuse their estimates and covariance using the EKF to obtain improved estimates for that particular time step. Note that despite the approximations involved, the computational complexity of these approaches remains $O(N^2)$ per measurement. Moreover, these approaches generate consistent estimates for the robots' poses.

Panzieri *et al.* [15] present an Interlaced EKF-based CL approach, where each robot processes only its own measurements to update its state, while considering the other robots' states as deterministic parameters. These state-covariance estimates are broadcast to robots within communication range. The computational and communication costs of this approach are $O(N)$ and $O(1)$ per measurement, respectively. In Martinelli's hierarchical EKF algorithm [11], the robot team is divided into L bounded-size groups, each of size M, such that $N = LM$. Each group has a leader that receives measurements from the robots in its group and processes them with computational cost $O(M^4)$ per time step. Furthermore, the group leaders can also form groups in a hierarchical manner, ensuring that the size of each group remains bounded. The main drawback of both [15] and [11], however, is that they ignore cross-correlations in order to reduce processing and communication costs, which may lead to overly optimistic and inconsistent estimates.

In the decentralized MLE-based CL by Howard *et al.* [6], each robot independently minimizes the part of the nonlinear cost function that contains terms involving its own state, considering the other robots' pose estimates as constants. When within communication range, robots exchange their state estimates. These estimates, received from other robots, are used to update the corresponding quantities in the local cost function. The main issue of concern, however, with this approach is that there exists no proof of its convergence.

Closely related to our proposed approach are the works of Arambel *et al.* [1] and Li and Nashashibi [10] that are based on the CI algorithm [7], which enables consistent fusion of estimates with unknown correlations. In [1], Arambel *et al.* develop a CI-based CL algorithm where each robot maintains the joint-state estimates of the entire team along with the corresponding $O(N^2)$ dense covariance matrix. Each robot locally processes its own relative measurements, using an EKF, to obtain updated joint-state estimates and covariance with processing cost $O(N^2)$ per measurement. The updated state-covariance estimates are then communicated to robots within communication range with $O(N^2)$ communication overhead. Robots fuse their local estimates with those received from other robots using CI.

In Li and Nashashibi [10], each robot maintains the decoupled state for the entire team. Specifically, each robot stores and updates only the 3×3 covariance matrix for every robot in the team, instead of the $O(N^2)$ dense covariance matrix for the joint state. On obtaining a relative measurement, the observer robot updates its local state-covariance using SCI [8] with processing cost $O(1)$. It then communicates its local information to other robots with cost $O(N)$, for fusion using the SCI again. This approach, however, ignores correlations while locally processing relative measurements using SCI (see Section III for details), which may lead to inconsistent and overly optimistic estimates.

The CI-based CL algorithm presented in this paper has two main advantages over [10] and [1]. First, as opposed to [10], our proposed algorithm is *provably consistent*. Second, as compared to [1], both its processing and communication requirements are only $O(1)$ per relative measurement, with a worst-case complexity of $O(N)$ per robot, per time step. We now present the details of our proposed algorithm.

III. Covariance-Intersection based Cooperative Localization

A. Problem Setup

Consider a team of N collaborating robots moving in 2D. We assume that each robot is equipped with (i) proprioceptive sensors (e.g., wheel encoders) that provide measurements of its ego-motion, (ii) exteroceptive sensors (e.g., camera, or laser scanner) that allow it to identify and obtain relative pose measurements to other robots in the team, and (iii) communication devices that permit information exchange between robots. Also, we assume that the robots' communication radius is larger than their sensing radius.

The state of the i-th robot (or R_i) at time-step k is denoted as $\mathbf{x}_k^i = [(\mathbf{p}_k^i)^T, \theta_k^i]^T$, where $\mathbf{p}_k^i = [x_k^i, y_k^i]^T$ and θ_k^i denote the robot's position and orientation, respectively, in the global frame of reference. The discrete-time motion model for R_i is given by:

$$\mathbf{x}^i_{k+1} = \mathbf{f}(\mathbf{x}^i_k, \mathbf{u}^i_{m,k}), \qquad i = 1, \ldots, N \tag{1}$$

where $\mathbf{f}$ is, in general, a nonlinear function, and $\mathbf{u}^i_{m,k} = \mathbf{u}^i_k + \mathbf{w}^i_k$. Here, $\mathbf{u}^i_k = [v^i_k, \omega^i_k]^T$ denotes the robot's true linear and angular velocity, which is corrupted by zero-mean, white Gaussian noise $\mathbf{w}^i_k = [w^i_{v,k}, w^i_{\omega,k}]^T$, with covariance $\mathbf{Q}^i_k$, to provide the corresponding measurement, $\mathbf{u}^i_{m,k}$.

The measurement model at time-step $k+1$, when R_i obtains a relative pose measurement, $\mathbf{z}^{i,j}_{k+1}$, to R_j, $i, j = 1, \ldots, N$, $j \neq i$, is given by:

$$\begin{aligned} \mathbf{z}^{i,j}_{k+1} &= \mathbf{h}^{i,j}(\mathbf{x}^i_{k+1}, \mathbf{x}^j_{k+1}) + \mathbf{n}^{i,j}_{k+1} & (2) \\ &= \Gamma^T_{\mathbf{x}^i_{k+1}}(\mathbf{x}^j_{k+1} - \mathbf{x}^i_{k+1}) + \mathbf{n}^{i,j}_{k+1} & (3) \end{aligned}$$

with

$$\begin{aligned} \Gamma_{\mathbf{x}^i_{k+1}} &= \begin{bmatrix} C(\theta^i_{k+1}) & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \\ C(\theta^i_{k+1}) &= \begin{bmatrix} \cos\theta^i_{k+1} & -\sin\theta^i_{k+1} \\ \sin\theta^i_{k+1} & \cos\theta^i_{k+1} \end{bmatrix} \end{aligned}$$

Here, $\mathbf{h}^{i,j}$ denotes the true relative pose between R_i and R_j at time-step $k+1$, and $\mathbf{n}^{i,j}_{k+1}$ is the zero-mean white Gaussian measurement noise with covariance, $\mathbf{R}^i_{k+1}$.

B. Algorithm Description

Consider all available measurement information up to time-step $k+1$, i.e., the velocity and relative pose observations, $\mathbf{u}^i_{m,K}$ and $\mathbf{z}^{i,j}_{K+1}$, $i, j = 1, \ldots, N$, $i \neq j$, $K = 0, \ldots, k$, respectively. In the standard EKF-based CL approach, each robot (or the fusion center) uses all the above information to calculate the joint-state estimates (of dimensions $3N \times 1$) of all robots, i.e., $\mathbf{x}_{k+1} = [\mathbf{x}^1_{k+1}{}^T, \mathbf{x}^2_{k+1}{}^T, \ldots, \mathbf{x}^N_{k+1}{}^T]^T$, at the current time-step $k+1$, along with the corresponding $3N \times 3N$ covariance matrix. Such an approach, however, incurs substantial computational, $O(N^4)$, and communication, $O(N)$, overhead per robot at each time step [16].

To address this problem, in our proposed approximate approach, each robot R_i locally maintains state and covariance estimates only for its *own* state $\mathbf{x}^i_{k+1}$. When R_i obtains a relative measurement, $\mathbf{z}^{i,j}_{k+1}$, to R_j, it generates a state-covariance estimate for the pose of R_j, using this relative measurement and its own state-covariance estimate. Next, R_i communicates this information to R_j who then fuses it with its own local state-covariance estimate using CI. We now present the mathematical details of this approach.

Using the motion model in (1), the state propagation equation for R_i is given by:[3]

$$\hat{\mathbf{x}}^i_{k+1|k} = \mathbf{f}(\hat{\mathbf{x}}^i_{k|k}, \mathbf{u}^i_{m,k}), \qquad i = 1, \ldots, N. \tag{4}$$

The corresponding covariance propagation equation for R_i is:

$$\mathbf{P}^i_{k+1|k} = \Phi^i_k \mathbf{P}^i_{k|k} \Phi^i_k{}^T + \mathbf{G}^i_k \mathbf{Q}^i_k \mathbf{G}^i_k{}^T \tag{5}$$

where $\Phi^i_k = \nabla_{\mathbf{x}^i_k}\mathbf{f}\Big|_{\mathbf{x}^i_k = \hat{\mathbf{x}}^i_{k|k}, \mathbf{w}^i_k = \mathbf{0}}$ and $\mathbf{G}^i_k = \nabla_{\mathbf{w}^i_k}\mathbf{f}\Big|_{\mathbf{x}^i_k = \hat{\mathbf{x}}^i_{k|k}, \mathbf{w}^i_k = \mathbf{0}}$. Since the dimensions of $\mathbf{x}^i_k$ are 3×1, from (4) – (5), we see that each robot R_i can propagate its own state-covariance estimate with processing cost $O(1)$ per time step.

When R_i obtains a relative pose measurement, $\mathbf{z}^{i,j}_{k+1}$, to R_j, R_i uses this measurement to generate an estimate, $\hat{\mathbf{x}}^{j*}_{k+1}$, for R_j, as follows [see (3)]:

$$\hat{\mathbf{x}}^{j*}_{k+1} = \hat{\mathbf{x}}^i_{k+1|k} + \Gamma_{\hat{\mathbf{x}}^i_{k+1|k}} \mathbf{z}^{i,j}_{k+1} \tag{6}$$

The error in this estimate is given by:

$$\tilde{\mathbf{x}}^{j*}_{k+1} = \tilde{\mathbf{H}}^{i,j}_{k+1} \tilde{\mathbf{x}}^i_{k+1|k} - \Gamma_{\hat{\mathbf{x}}^i_{k+1|k}} \mathbf{n}^{i,j}_{k+1} \tag{7}$$

where

$$\tilde{\mathbf{H}}^{i,j}_{k+1} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(\hat{\mathbf{p}}^{j*}_{k+1} - \hat{\mathbf{p}}^i_{k+1|k}) \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \quad , \quad \mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix},$$

and $\mathbf{I}_2$ is the identity matrix. Note that $\tilde{\mathbf{H}}^{i,j}_{k+1} = \nabla_{\mathbf{x}_k}\mathbf{h}^{i,j}\big|_{\mathbf{x}^i_k = \hat{\mathbf{x}}^i_{k+1|k}, \mathbf{x}^j_k = \hat{\mathbf{x}}^{j*}_{k+1}}$. Therefore, the corresponding covariance is calculated as:

$$\mathbf{P}^{j*}_{k+1} = \tilde{\mathbf{H}}^{i,j}_{k+1} \mathbf{P}^i_{k+1|k} \tilde{\mathbf{H}}^{i,j^T}_{k+1} + \Gamma_{\hat{\mathbf{x}}^i_{k+1|k}} \mathbf{R}^i_{k+1} \Gamma^T_{\hat{\mathbf{x}}^i_{k+1|k}} \tag{8}$$

We make three main observations regarding this step: (i) the Jacobians in (8) can be completely evaluated using R_i's *locally-available* estimates, $\hat{\mathbf{x}}^i_{k+1|k}$ and $\hat{\mathbf{x}}^{j*}_{k+1}$, respectively, (ii) the processing cost is $O(1)$, and (iii) even though each robot R_i does not include R_j's state in its local state vector, it can generate an estimate for R_j every time it obtains a relative measurement, $\mathbf{z}^{i,j}_{k+1}$, to R_j.

After R_i computes a state-covariance estimate for R_j [see (6) and (8)], it communicates this information to R_j with communication cost $O(1)$. Next, R_j fuses $\hat{\mathbf{x}}^{j*}_{k+1}$ and $\mathbf{P}^{j*}_{k+1}$ with its own local state-covariance estimate, $\hat{\mathbf{x}}^j_{k+1|k}$ and $\mathbf{P}^j_{k+1|k}$, to obtain improved estimates for itself. Here, it is important to note that the estimates $\hat{\mathbf{x}}^{j*}_{k+1}$ and $\hat{\mathbf{x}}^j_{k+1|k}$, for the random variable, $\mathbf{x}^j_{k+1}$, are *not independent*. This is because, if robots R_i and R_j have exchanged information either *directly or indirectly*, through other robots, at *any* previous time-step m, $m \leq k$, then their estimates at time-step $k+1$ are correlated. Since the robots do not keep track of these cross-correlation terms in our approach, the correlations between these estimates is unknown. Therefore, the commonly used EKF update step [12], designed for fusing independent estimates, cannot be employed here as it may lead to estimator inconsistency due to information reuse.

To address this problem, we fuse these estimates using the CI algorithm. Presented in [7], the CI is designed to *consistently* fuse estimates with unknown correlations (see

[3] $\hat{\mathbf{y}}_{l|m}$ denotes the estimate of the random varaible $\mathbf{y}$ at time-step l, given measurement information up to time-step m.

Appendix A). Using CI, the covariance update step in our algorithm is given by:

$$\mathbf{P}^{j}_{k+1|k+1} = \left[\omega(\mathbf{P}^{j}_{k+1|k})^{-1} + (1-\omega)(\mathbf{P}^{j*}_{k+1})^{-1}\right]^{-1} \quad (9)$$

where $\omega \in [0,1]$ is chosen so that the trace of $\mathbf{P}^{j}_{k+1|k+1}$ is minimized.

The updated pose estimate, $\hat{\mathbf{x}}^{j}_{k+1|k+1}$, is computed as follows:

$$\begin{aligned}\hat{\mathbf{x}}^{j}_{k+1|k+1} &= \mathbf{P}^{j}_{k+1|k+1}\Big[\omega(\mathbf{P}^{j}_{k+1|k})^{-1}\hat{\mathbf{x}}^{j}_{k+1|k} \\ &\quad +(1-\omega)(\mathbf{P}^{j*}_{k+1})^{-1}\hat{\mathbf{x}}^{j*}_{k+1}\Big] \qquad (10)\end{aligned}$$

Importantly, both $\hat{\mathbf{x}}^{j}_{k+1|k+1}$ and $\mathbf{P}^{j}_{k+1|k+1}$ can be calculated by R_j with processing cost $O(1)$. Therefore, both the overall processing and communication requirements of our proposed consistent algorithm remain $O(N)$ per robot per time step. As a final remark, we note that the two terms in the covariance update [see (8)] are not independent, since they both depend upon the estimate, $\hat{\mathbf{x}}^{i}_{k+1|k}$, of robot R_i. Therefore, they cannot be fused using the SCI algorithm, as claimed in [10].

IV. Simulation and Experimental Results

A. Simulations

The performance of the proposed decentralized CI-based CL algorithm was tested in simulation with a team of $N=3$ robots in 2D, following circular non-concentric trajectories. The robots move in an area of approximately 25 m×30 m for 6000 time steps (each time step has a duration of 0.01 sec). Each robot measures its linear, v_m, and rotational, ω_m, velocity, as well as the relative pose, $\mathbf{z} = [\Delta x, \Delta y, \Delta\theta]^T$, [as described in (3)] of other robots in the team. The measurement noise is modeled as zero-mean white Gaussian with standard deviation of $\sigma_{v_m} = 2\%v$, $\sigma_{\omega_m} = 1$ deg/sec for the linear and rotational velocity, respectively, and $\sigma_{\Delta x} = 0.05$ m, $\sigma_{\Delta y} = 0.05$ m, and $\sigma_{\Delta\theta} = 1$ deg for the corresponding relative measurements. Each robot propagates its own state at every time step, while the measurement update is carried out every 5 time steps. The criterion established for robot detection is based on the relative distance, d, which in this case is set to $d \leq 10$ m.

We compared the performance of the following CL approaches: (i) EKF-CL (processing and communication complexity, $O(N^4)$ and $O(N)$, respectively, per robot at each time step), (ii) Proposed CI-CL (both processing and communication complexity $O(N)$ per robot at each time step), and (iii) SCI-CL [10] (both processing and communication complexity, $O(N)$, per robot per time step).

We employ the root mean square error (RMSE) criterion to test the accuracy of these 3 approaches. Fig. 1 shows the RMSE in the robot's position estimates (averaged over 50 runs) at each time step. As expected, the EKF-CL, which is our benchmark, is the most accurate. This is due to the fact that it maintains the joint-state for the robots, along with the corresponding dense covariance matrix that keeps track of the cross-correlations between the state estimates. Our proposed CI-CL approach, is less accurate than EKF-CL, since it does not compute the cross-correlations between the robots' state estimates. However, it reduces the computational cost and guarantees a consistent result, while sacrificing estimation accuracy. Finally, the SCI-CL method [10], which has the same processing and communication requirements as our proposed CI-CL approach, is the least accurate. This can be attributed to the fact that the SCI-CL performs two approximations (SCI updates), resulting in additional loss of information.

Fig. 2, that depicts the RMSE in the orientation estimates, corroborates the results of Fig. 1. As is expected, the EKF-CL is the most accurate, followed by our CI-CL implementation which outperforms the SCI-CL. Furthermore, Fig. 3 shows the sum of the standard deviations in the robots' position estimates. As we can see, EKF-CL has the lowest standard deviation, followed by our CI-CL, while the SCI-CL is the most conservative. As stated above, the loss of information in the SCI-CL can be attributed to the additional, artificial noise that is injected in the propagation model in [10], along with the successive application of two SCI updates. Finally, Fig. 4 shows the Normalized Estimation Error Squared (NEES) for the 3 approaches, averaged over 50 runs. From the figure we see that the NEES for the EKF-CL, that maintains the joint-state is around the ideal value of 9. For the proposed CI-CL, the NEES for each robot is around the ideal value of 3. This indicates that both these approaches are consistent. The NEES for SCI-CL is higher than the ideal value of 3 for each robot, indicating that this algorithm is inconsistent.

B. Experiments

The performance of the proposed decentralized CI-CL algorithm was also tested with real hardware. We used a dataset consisting of the odometry measurements recorded by 4 Pioneer 3DX robots moving in a 2D environment, while an overhead camera computes their ground truth. The robots move in an area of approximately 2.5 m×4 m for 950 time steps (each time step has a duration of 1 sec). The relative pose measurements are computed synthetically using the ground truth corrupted by zero-mean white Gaussian noise with standard deviations $\sigma_{\Delta x} = 0.05$ m, $\sigma_{\Delta y} = 0.05$ m, and $\sigma_{\Delta\theta} = 1$ deg. The sensing radius for the robots is set to 2 m.

Figs. 5 and 6 show the error in robots' position and orientation respectively, during the experiment. As evident, the performance of our proposed approach, with only linear processing and communication requirements, is very close to that of the EKF-CL, which has quadratic processing and linear communication requirements. Moreover, our CI-CL approach clearly outperforms the competing linear-time solution of SCI-CL.

V. Conclusion and Future Work

In this paper, we presented an approximate decentralized algorithm for CL based on the CI algorithm. The proposed approach, where each robot only maintains an estimate of its own state and covariance, reduces both the

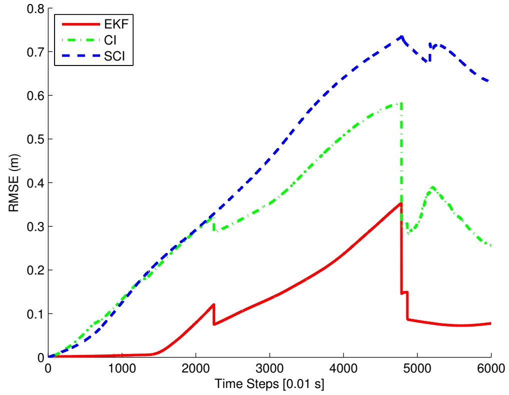

Fig. 1. Simulation results: RMSE in the robots' position estimates.

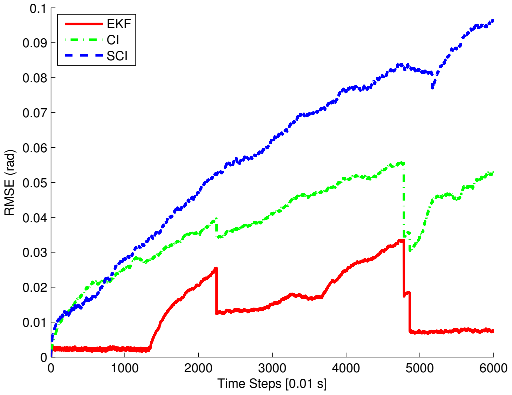

Fig. 2. Simulation results: RMSE in the robots' orientation estimates.

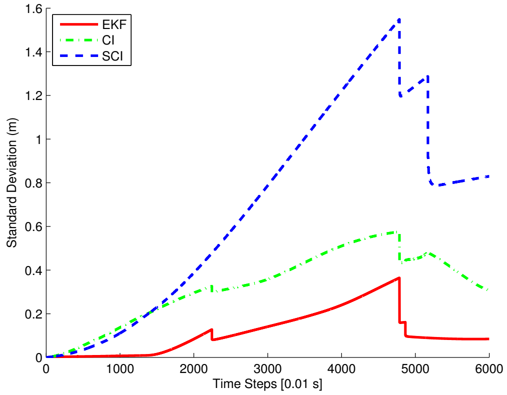

Fig. 3. Simulation results: Standard deviation in the robots' position estimates.

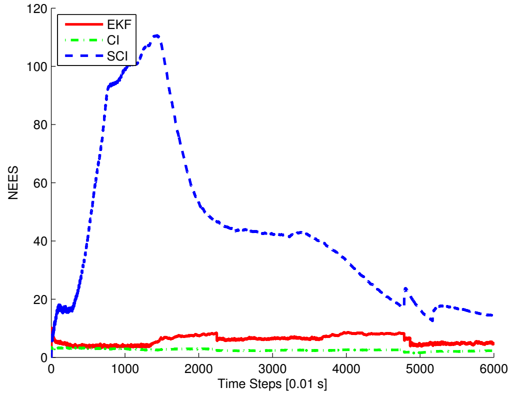

Fig. 4. Simulation results: Average NEES.

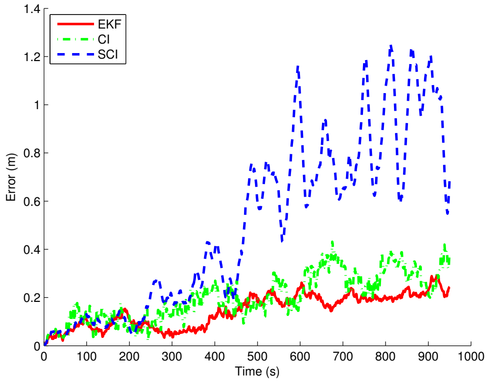

Fig. 5. Experimental results: error in the robots' position estimates.

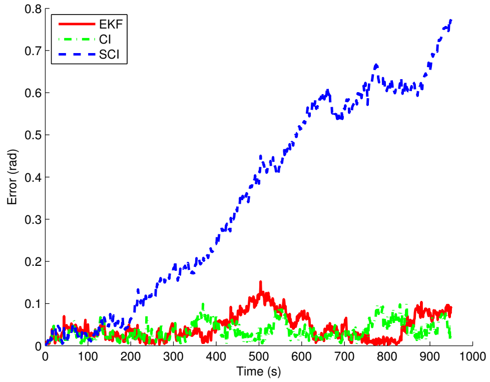

Fig. 6. Experimental results: error in the robots' orientation estimates.

processing and communication costs of CL to $O(1)$ per relative measurement, by avoiding the costly computation of the cross-correlation terms between the robots. Moreover, our approach is consistent and can efficiently handle asynchronous communication constraints. Lastly, simulation and experimental results have validated the performance and accuracy of our proposed algorithm and shown that it outperforms alternative approximate CL methods of comparable processing and communication costs. As part of our future work, we will extend our proposed approach to robot teams navigating in 3D using visual and inertial measurements.

Appendix

A. Review of Covariance Intersection

Consider two consistent estimates, $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$, of a quantity of interest, $\mathbf{x}$, with corresponding covariances and cross-correlation terms given by $\mathbf{P}_{aa}$, $\mathbf{P}_{bb}$, and $\mathbf{P}_{ab}$ respectively. These estimates can be combined to obtain an improved consistent estimate $\hat{\mathbf{c}}$ with covariance $\mathbf{P}_{cc}$ as follows:

$$\begin{aligned} \hat{\mathbf{c}} &= \mathbf{W}_a\hat{\mathbf{a}} + \mathbf{W}_b\hat{\mathbf{b}} \\ \mathbf{P}_{cc} &= \mathbf{W}_a\mathbf{P}_{aa}\mathbf{W}_a^T + \mathbf{W}_a\mathbf{P}_{ab}\mathbf{W}_b^T + \mathbf{W}_b\mathbf{P}_{ba}\mathbf{W}_a^T + \mathbf{W}_b\mathbf{P}_{bb}\mathbf{W}_b^T \end{aligned} \tag{11}$$

where $\mathbf{W}_a$ and $\mathbf{W}_b$ are weights chosen to minimize the trace of $\mathbf{P}_{cc}$. Note that when these two estimates are independent, i.e., $\mathbf{P}_{ab} = \mathbf{0}$, this fusion step becomes identical to the conventional Kalman filter update [12]. However, when $\mathbf{P}_{ab} \neq \mathbf{0}$, if the conventional Kalman filter equations are used, which ignore this cross-covariance term, information will be re-used, resulting in overly optimistic and inconsistent estimates.

To address this problem, Julier and Uhlmann introduced the Covariance Intersection (CI) algorithm in [7] for fusing estimates (of the same quantity) with unknown correlations. CI uses a convex combination of the mean and covariance estimates ($\{\hat{\mathbf{a}}, \mathbf{P}_{aa}\}$ and $\{\hat{\mathbf{b}}, \mathbf{P}_{bb}\}$) to obtain a new estimate $\hat{\mathbf{c}}^*$ with covariance $\mathbf{P}_{cc}^*$, which is guaranteed to be consistent, as follows:

$$\mathbf{P}_{cc}^* = \left(\omega\mathbf{P}_{aa}^{-1} + (1-\omega)\mathbf{P}_{bb}^{-1}\right)^{-1} \tag{12}$$

$$\hat{\mathbf{c}}^* = \mathbf{P}_{cc}^*\left(\omega\mathbf{P}_{aa}^{-1}\hat{\mathbf{a}} + (1-\omega)\mathbf{P}_{bb}^{-1}\hat{\mathbf{b}}\right) \tag{13}$$

where $\omega \in [0,1]$ is a parameter chosen to minimize either the trace or determinant of $\mathbf{P}_{cc}$.

The intuition behind CI arises from a geometric interpretation of (11), based on which, $\mathbf{P}_{cc}$ will always lie in the intersection of $\mathbf{P}_{aa}$, $\mathbf{P}_{bb}$ for any value of $\mathbf{P}_{ab}$. Fig. 7 shows this for different values of $\mathbf{P}_{ab}$. This interpretation led to the development of an updating strategy (i.e., the CI algorithm) that finds the $\mathbf{P}_{cc}$, which encloses the intersection region of $\mathbf{P}_{aa}$ and $\mathbf{P}_{bb}$ ensures consistency without the need of knowing the value of $\mathbf{P}_{ab}$. Fig. 8 illustrates this strategy for different values of ω.

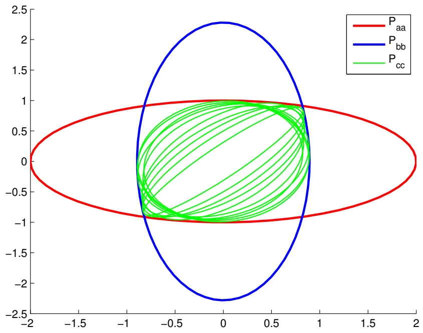

Fig. 7. Covariance ellipsoids for $\mathbf{P}_{aa}$,$\mathbf{P}_{bb}$, and the updated $\mathbf{P}_{cc}$ estimate for different values of $\mathbf{P}_{ab}$.

References

[1] P.O. Arambel, C. Rago, and R.K. Mehra, "Covariance intersection algorithm for distributed spacecraft state estimation", in *Proc. of the IEEE American Control Conference*, Arlington, VA, Jun. 25-27, 2001, pp. 4398-4403.

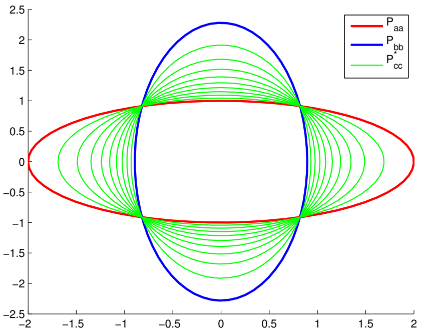

Fig. 8. Covariance ellipsoids for $\mathbf{P}_{aa}$,$\mathbf{P}_{bb}$, and the CI updated $\mathbf{P}_{cc}^*$ estimate for different values of ω.

[2] T. Bailey, M. Bryson, H. Mu, J. Vial, L. McCalman, and H. Durrant-Whyte, "Decentralised cooperative localisation for heterogeneous teams of mobile robots", in *Proc. of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9-13, 2011, pp. 2859-2865.

[3] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*, Wiley, New York, NY, 2001.

[4] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization", *Autonomous Robots*, vol. 8, no. 3, pp. 325-344, Jun. 2000.

[5] A. Howard, M.J. Matarić, and G.S. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation", in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, EPFL, Switzerland, Sep. 30 - Oct. 5, 2002, pp. 434-439.

[6] A. Howard, M.J. Matarić, and G.S. Sukhatme, "Localization for mobile robot teams: a distributed MLE approach", in *Experimental Robotics VIII*, Springer-Verlag, 2003, vol. 5, pp. 146-155.

[7] S.J. Julier and J.K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations", in *Proc. of the IEEE American Control Conference*, Albuquerque, NM, Jun. 4-6, 1997, pp. 2369-2373.

[8] S.J. Julier and J.K. Uhlmann, "General decentralized data fusion with covariance intersection (CI)", in *Handbook of Data Fusion*, CRC Press, Boca Raton, FL, 2001, pp. 272-276.

[9] N. Karam, F. Chausse, R. Aufrere, and R. Chapuis, "Localization of a group of communicating vehicles by state exchange", in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 9-15, 2006, pp. 519-524.

[10] H. Li and F. Nashashibi, "Cooperative Multi-Vehicle Localization Using Split Covariance Intersection Filter", in *Proc. of the IEEE Intelligent Vehicles Symposium*, Alcalá de Henares, Spain, Jun. 3-7, 2012, pp. 211-216.

[11] A. Martinelli, "Improving the precision on multi robot localization by using a series of filters hierarchically distributed", in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, Oct. 29 - Nov. 2, 2007, pp. 1053-1058.

[12] P. Maybeck, *Stochastic Models, Estimation and Control*, Academic Press, New York, NY, 1979, vol. 1.

[13] A.I. Mourikis and S.I. Roumeliotis, "Optimal sensor scheduling for resource-constrained localization of mobile robot formations", in *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 917-931, Oct. 2006.

[14] E.D. Nerurkar, S.I. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization", in *Proc. of the IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 12-17, 2009, pp. 1402-1409.

[15] S. Panzieri, F. Pascucci, and R. Setola, "Multirobot localization using interlaced extended Kalman filter", in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, Oct. 9-15, 2006, pp. 2816-2821.

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