J Intell Robot Syst (2012) 66:321–342
DOI 10.1007/s10846-011-9620-2

Decentralized Cooperative SLAM for Sparsely-Communicating Robot Networks: A Centralized-Equivalent Approach

Keith Y. K. Leung · Timothy D. Barfoot · Hugh H. T. Liu

Received: 14 December 2010 / Accepted: 13 July 2011 / Published online: 18 August 2011

Abstract Communication between robots is key to performance in cooperative multi-robot systems. In practice, communication connections for information exchange between all robots are not always guaranteed, which adds difficulty in performing state estimation. This paper examines the decentralized cooperative *simultaneous localization and mapping (SLAM)* problem, in which each robot is required to estimate the map and all robot states under a sparsely-communicating and dynamic network. We show how the exact, centralized-equivalent estimate can be obtained by all robots in the network in a decentralized manner even when the network is never fully connected. Furthermore, a robot only needs to consider its own knowledge of the network topology in order to detect when the centralized-equivalent estimate is obtainable. Our approach is validated through more than 250 min of hardware experiments using a team of real robots. The resulting estimates are compared against accurate groundtruth data for all robot poses and landmark positions. In addition, we examined the effects of communication range limit on our algorithm's performance.

Keywords Networked robots · Decentralized state estimation · Finite sensing and communication · SLAM · Autonomous agents

K. Y. K. Leung (✉) · T. D. Barfoot · H. H. T. Liu
University of Toronto Institute for Aerospace Studies,
4925 Dufferin St., Toronto, ON, M3H 5T6, Canada
e-mail: keith.leung@robotics.utias.utoronto.ca

T. D. Barfoot
e-mail: tim.barfoot@utoronto.ca

H. H. T. Liu
e-mail: liu@utias.utoronto.ca

1 Introduction

A cooperative multi-robot system is beneficial in many applications. It allows for the implementation of complex strategies that require more than a single robot. Multiple robots can also provide a certain degree of redundancy to ensure the completion of tasks should a portion of the multi-robot team become disabled. Communication and the mutual exchange of information are key performance factors for many cooperative multi-robot systems. However, the limitation of communication range and its impact on a multi-robot system have only occasionally been the focus of research.

In this paper, we examine the cooperative decentralized *simultaneous localization and mapping (SLAM)* problem, in which we require each robot to estimate the map (landmarks) and the state of all robots in a sparsely-communicating and dynamic network. This implies that the communication network is never guaranteed to be fully connected at any time. Such scenarios may occur in applications where robots are operating over a large area, or in places where occlusions limit the communication range.

Our novel contributions are as follows:

1. We present a decentralized cooperative SLAM algorithm that allows each robot to obtain the exact centralized-equivalent estimate for all robots and known landmarks whenever possible, even if the robot network is never fully connected. This algorithm also ensures that a robot only needs to consider its own knowledge before applying the Markov Property to discard information that it no longer needs, while ensuring that all other robots can still obtain the centralized-equivalent estimate. In other words, a robot does not need to keep track of what other robots know.
2. We performed extensive hardware experiments, the results of which are used to validate our approach. We have performed over 250 min of hardware experiments using a fleet of five robots (as shown in Fig. 1), where we have accurate groundtruth data of all robot poses and all landmark positions (which is rarely available for SLAM experiments) for all times.

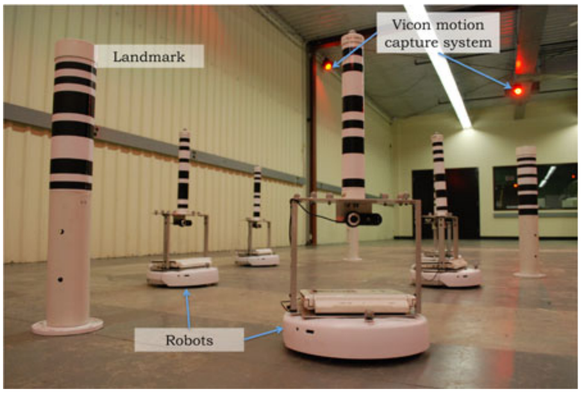


Fig. 1 Our decentralized cooperative SLAM experiment uses a fleet of five robots built using the iRobot Create platform. A Vicon motion capture system consisting of 10 cameras provides accurate groundtruth information for both robots and landmarks during our experiments

3. We provide a comprehensive analysis of how network connectivity affects state estimation performance, and show that a robot's decentralized estimate of the map and its own pose always remains close to the centralized-equivalent estimate even when robots rarely communicate with each other.

As we shall see in our review of related work in Section 2, there have been various definitions of the cooperative SLAM problem. In this paper, we specifically require that each robot obtains estimates for all robot poses and the positions for all known landmarks. In particular, we want each robot to obtain the exact centralized-equivalent estimate whenever possible. This is an estimate that is mathematically equivalent to the centralized estimate, but obtained in a decentralized manner. One important attribute of the centralized-equivalent estimate is that it accounts for all measurements in the order in which they were made. Furthermore, the exact centralized-equivalent estimate does not suffer from estimator consistency problems caused specifically by cyclic updates, and the merging of state estimates to produce approximations to the centralized estimate. Essentially, the centralized-equivalent estimate is the best estimate that can be produced online. However, due to operating in a network that is never guaranteed to be fully connected, the centralized-equivalent estimates may be delayed in time as robots wait for the required information to arrive over the network. In this situation, our approach allows robots to use the currently available information to calculate a temporary state estimate (which can be replaced by the centralized-equivalent estimate later).

We will begin with a review of some related work in multi-robot state estimation in Section 2. Section 3 contains the formulation for our decentralized cooperative SLAM problem. In Section 4, we present the theoretical work that forms the basis of our proposed algorithm, and also discuss the necessary initial conditions for the decentralized cooperative SLAM problem. In Section 5, we present our decentralized algorithm and its complexity analysis. Section 6 describes our experimental setup, and the experimental results are presented in Section 7.

2 Related Work

Cooperative SLAM is closely related to the problem of *cooperative localization*, and performing state estimation within sensor networks. The cooperative localization problem, however, is relatively easier to solve in that the map does not need to be estimated. Instead, robots only need to estimate the poses of all robots by using relative measurements between robots as well as odometry information. Kurazume and Hirose [1] introduced one of the earliest methods for cooperative localization. In their approach, robots are split into two groups. One group of robots moves while the other remains stationary, essentially serving as landmarks for the first group of robots for localization. Roumeliotis and Bekey [2] performed distributed cooperative localization by decomposing the Extended Kalman filter (EKF) into a number of filters that can perform the prediction step of the EKF locally on each robot. This is done by factoring the covariance matrix using singular value decomposition, such that each robot can individually calculate the effects of their motion update on the covariance matrix. The distributed calculations are combined before a measurement update, which then requires full network connectivity (i.e., all robots need to communicate with one another for the EKF correction step). Later, Roumeliotis and Rekleitis [3] analytically quantified the benefit of cooperative localization. Their work shows how the number of robots in a team can influence localization performance. In particular, their work shows that there are diminishing returns in terms of localization performance (uncertainty reduction) with an increase in the number of robots.

An algorithm for performing decentralized cooperative localization in sparsely-communicating robot networks was introduced in [4]. Some of the concepts in this paper are partially derived from this work. However, to distinguish our current contributions, this paper examines the more complicated decentralized cooperative SLAM problem, where landmark positions need to be estimated along with the poses of all robots. We also demonstrate the validity of our approach using extensive real hardware experiments (instead of only using simulations). The work in this paper can be viewed as a generalization of [4]. Based on this past work, Nerurkar and Roumeliotis [5] recently examined different communication strategies when robots must operate with bandwidth constraints on the network.

Some of the preliminary results of this paper are presented in [6], but this paper adds significantly more insight into our decentralized cooperative SLAM algorithm, and a more thorough discussion on memory use, computational complexity, and communication complexity. Additionally, we provide a detailed analysis on the effect of communication range limit on the performance of our algorithm.

As we will see in the development of our proposed algorithm, the centralized-equivalent estimates that we want all robots to obtain can be delayed due to the absence of a fully-connected network, which prevents the communication of information (measurements) between robots. As the network evolves, robots may also receive *out-of-sequence measurements* (OOSM). Bar-Shalom [7] and Bar-Shalom et al. [8] examined some possible remedies for treating OOSM in state estimation using a Kalman filter (also known as the negative timestep problem), and applied this to target tracking. His work shows that the only way to incorporate a missed measurement and still produce a centralized state estimate is to sequentially reprocess all following measurements. This is an important concept to note for the rest of this paper as it dictates the information that a robot needs to retain to calculate the centralized-equivalent estimate. Furthermore, it helps in understanding the computational complexity of our proposed algorithm. A concept similar to this is used by Capitán et al. [9] in their *delayed-state* filtering approach, where the entire trajectory of a robot is stored, and truncated when its length reaches a threshold (memory limit). In contrast, our approach applies the Markov property to limit memory use. Furthermore, contrary to [10], we do not require a robot to know what other robots know before applying the Markov property, while still guaranteeing that all robots can obtain the centralized-equivalent estimate.

As network properties can cause difficulties by preventing the exchange of information, robots may also experience the problem of over-using the same piece of information. This problem, known as *cyclic update*, occurs when one robot provides information to another robot to update its state estimate, which in turn provides merged information back to the first robot for updating its state estimate (i.e., information is double counted). The results of this are inconsistent, and overconfident state estimates. Howard et al. [11] introduced the *dependency tree* as a remedy to this problem for a multi-robot system but it is not guaranteed to always work. The decentralized state estimation method that we present in this paper does not suffer from cyclic updates. This is because we always seek to obtain the centralized-equivalent estimate, which properly accounts for all measurements only once, and in proper sequence.

Before we look at some additional solutions to cooperative SLAM, it is important to note that this problem has been defined differently in various works. In this paper, we require each robot to obtain estimates for the position of all known landmarks, as well as the poses of all robots in the system. Fenwick et al. [12] generalized the single-robot *Extended Kalman Filter* (EKF) SLAM approach for the multi-robot scenario. This is essentially a centralized approach, which requires a fully-connected communication network. Using the information form of the EKF, Nettleton et al. [13] devised a decentralized algorithm that takes advantage of the additive update property, which allows the incorporation of observations into an estimate in any order. However, as pointed out by Rosencrantz et al. [14], this only works for states that do not change over time and where motion predictions are not necessary (i.e., only the static map is estimated). Nettleton et al. [15] acknowledges this, because when we estimate the states of robots and landmarks, motion predictions affect the estimate uncertainty for the entire system. Nettleton et al. [15] also presents a SLAM algorithm where robots only communicate sub-map information to reduce communication bandwidth. However, this no longer produces centralized-equivalent estimates. Furthermore, as Reece and Roberts [16] point out, the use of the Covariance Intersection method as part the algorithm in [15] can yield highly conservative estimates. In addition, the algorithm can only handle Gaussian noise as it is fundamentally based on the *Extended Information Filter*. The work presented in this paper does not have this limitation.

Thrun et al. [17] introduced the *Sparse Extended Information Filter (SEIF)* for SLAM, which is an approximation of the centralized-equivalent estimate. This was extended to the multi-robot SLAM problem, where the robots are not aware of each other's starting pose, and overlapping local maps from each robot are combined to form the global map. Similarly, Ko et al. [18], Zhou and Roumeliotis [19] and Wang et al. [20] also came up with approaches that involve merging of maps between robots. In the approach proposed in [18], a robot tries to localize itself in another robot's map to confirm their relative positions before merging their maps. In [19], relative measurements between robots, as well as landmarks in two robots' maps are used to align and merge maps. Note that although the approaches described above do not require a fully-connected network at all times, they do not produce centralized-equivalent estimates. In contrast, our approach allows robots to obtain the centralized-equivalent estimate whenever possible, even without a fully-connected network.

Howard et al. [21] presented a novel approach to multi-robot SLAM that uses manifold maps instead of a planar map for a two dimensional environment. In his approach. each robot maintains its own manifold map until it encounters another robot, at which point their maps are merged. The focus of this work is mainly on the mapping aspect, and a robot only needs to localize itself (as opposed to localizing all robots in the system in our current work). Howard [22] also looked at performing multi-robot SLAM using particle filters. In this approach, robots are unaware of each other's initial pose and first perform state estimation in a decentralized manner (i.e., each robot acts as an independent system). When robots encounter each other for the first time, their individual maps are combined into a common map using relative pose information. The notion of a virtual robot traveling backward in time is introduced to allow the incorporation of information gathered by a robot before the common map is merged. When all robots have encountered each other, the state estimation process becomes centralized, and requires a fully-connected communication network. To reiterate, this connectivity requirement is not necessary in our work.

In the following sections, we will present an algorithm that is able to provide the centralized-equivalent estimate of all robots and observed landmarks whenever possible in a sparsely-communicating and dynamic network that never has to be fully connected. Our algorithm applies the Markov property to limit the amount of information that needs to be stored, but at the same time, does not require a robot to know what other robots know to ensure that all robots can still obtain the centralized-equivalent estimate.

3 Problem Formulation

In a multi-robot system, let N represent the set containing the unique identification indices of all robots. In addition to this, let M represent the set that contains the unique identification indices of all landmarks (the map). The system model is as follows:

$$\mathbf{x}_{i,k} = \mathbf{g}\left(\mathbf{x}_{i,k-1}, \mathbf{u}_{i,k}, \epsilon_k\right), \ (\forall i \in N)$$

$$\mathbf{x}_{m,k} = \mathbf{x}_{m,k}, \ (\forall m \in M)$$

$$\mathbf{y}_{i,k}^{j,i} = \mathbf{h}\left(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}, \mathbf{x}_{m,k}, \delta_k\right), (\forall j \in N)\left(d_k^{j,i} \leq r_{\mathsf{obs}}\right)$$

$$\mathbf{y}_{i,k}^{m,i} = \mathbf{h}\left(\mathbf{x}_{i,k}, \mathbf{x}_{m,k}, \delta_k\right), (\forall m \in M)\left(d_k^{m,i} \leq r_{\mathsf{obs}}\right)$$

where for timestep k:

- $\mathbf{x}_{i,k}$ represents the state (pose) of robot i
- $\mathbf{x}_{m,k}$ represents the state (position) of the stationary landmark m
- $\mathbf{u}_{i,k}$ represents the odometry information of robot i
- $\mathbf{g}(\cdot)$ is the state transition function for the robots
- ϵ_k represents the process noise
- $\mathbf{y}_{i,k}^{j,i}$ represents the measurement (e.g., range/bearing) of robot j with respect to robot i
- $\mathbf{y}_{i,k}^{m,i}$ represents the measurement of landmark m with respect to robot i
- $\mathbf{h}(\cdot)$ is the measurement function
- δ_k is the measurement noise
- $d_k^{j,i}$ is the distance between robot i and robot j
- $d_k^{m,i}$ is the distance between robot i and landmark m
- r_{obs} is the measurement range limit

Let

$$X_k = \left\{\mathbf{x}_{i,k}, \mathbf{x}_{m,k} | i \in N, m \in M_k\right\},$$

represent the set of all states known to exist at timestep k, where M_k is the set of landmarks that has been observed by at least one robot up to time k. Let

$$X_{i,k} = \left\{\mathbf{x}_{j,k}, \mathbf{x}_{m,k} | j \in N_{i,k}, m \in M_{i,k}\right\},$$

represent the states of all robots and landmarks known to robot i up to time k, where $N_{i,k}$ is the set of robots known to robot i up to time k, and $M_{i,k}$ is the set of landmarks known to robot i up to time k. Let

$$Y_{i,k} = \left\{\mathbf{y}_{i,k}^{j,i}, \mathbf{y}_{i,k}^{m,i} | j \in N, m \in M, \right.$$
$$\left. d_k^{j,i} \leq r_{\mathsf{obs}}, d_k^{m,i} \leq r_{\mathsf{obs}}\right\}$$

represent the set of measurements from robot i to all robots and landmarks within observation range. Finally, let

$$R_{i,k} = \left\{j | j \in N, d_k^{i,j} \leq r_{\mathsf{comm}}\right\}$$

represent the set of robots that can exchange information with robot i (either directly or by relaying information through another robot) at time k. In other words, this is the set of robots connected to robot i.

Due to uncertainty in both state transition and measurements, the true state of the system cannot be found deterministically, but can only be estimated using odometry and measurement data. In general, the centralized *belief*, $\mathrm{bel}(X_k)$, is represented by a probability density function, $p(\cdot)$, over all states, X_k:

$$\mathsf{bel}(X_k) := p\left(X_k | \mathsf{bel}(X_0), \left\{\mathbf{u}_{i,1:k}\right\}, \left\{Y_{i,1:k}\right\}, (\forall i)\right),$$

which is conditioned on the initial belief, $\mathsf{bel}(X_0)$, past odometry data, and past range and bearing measurements.

The *knowledge set*, $S_{i,k}$, consists of all odometry and measurement data, as well as the previous state estimates known to robot i at time k. We assume at the initial timestep that each robot has an estimate of its pose,

$$S_{i,0} = \left\{\mathsf{bel}(\mathbf{x}_{i,0})\right\}.$$

We will also assume that each robot initially knows the total number of robots in the team. This is an important requirement, which we will discuss in greater detail later in Section 5. Robots within communication range r_{comm} of each other are able to exchange and relay state estimates, odometry data, and measurement data. Let $S^-_{i,k}$ represent the knowledge set after state transition and observations, but before communication is established with any other robot:

$$S^-_{i,k} = S_{i,k-1} \cup \left\{\mathbf{u}_{i,k}, Y_{i,k}\right\} \tag{1}$$

When communication occurs between robots i and j, they will make their knowledge sets available to each other as follows, where the $\underset{j\in R_{i,k}}{\cup}$ operator represents the union with all the sets indexed by j:

$$S_{i,k} = S^-_{i,k} \underset{j\in R_{i,k}}{\cup} S^-_{j,k} \tag{2}$$

From a practical and computation point of view, it is helpful to apply the *Markov property*,

$$p\left(X_k|\mathsf{bel}(X_0), U_{1:k}, Y_{1:k}\right)$$
$$= p\left(X_k|\mathsf{bel}(X_{k-1}), U_k, Y_k\right),$$

when performing state estimation, as it limits memory and processing requirements and allows for recursive state estimation. The difficulty is that in a decentralized framework, the Markov property can only be applied once a robot obtains sufficient information regarding other robots through communication. Furthermore, each robot must ensure that other robots will no longer require any of the past information that will be discarded when applying the Markov property. Our objective is for each robot to perform decentralized cooperative SLAM to estimate the state of all robots and known landmarks (i.e., find $\mathsf{bel}(X_{i,k})$) in a decentralized manner, given that the robots are only occasionally exchanging information with one another in a sparsely-connected and dynamic network.

4 Decentralized Cooperative SLAM in a Dynamic Network

In developing a solution to our decentralized cooperative SLAM problem, we will first look at the problem from the perspective of an outside observer, who has the benefit of seeing which robots are in communication at every timestep. We will define the event that signifies that the centralized-equivalent estimate is obtainable for all robots, and see how it is a generalization of the same event in the robot-only cooperative localization problem. Realistically, robots within the network do not have the benefit of knowing if other robots are in communication at a particular timestep. Hence, we will examine the evolution of a network from a robot's local perspective and see how a robot can obtain the centralized-equivalent estimate. Again, we will see how this is generalized from the cooperative localization case. Finally, we will look at the initial conditions that must be imposed on the network of robots.

4.1 Obtaining the Centralized-Equivalent Estimate—The Global Perspective

An outside observer of the system is able to see when each robot makes a measurement or communicates with another robot. We will begin by looking at the necessary and sufficient conditions for all robots to be able to obtain the centralized-equivalent estimate. In decentralized cooperative localization [4], the necessary and conditions are satisfied when a checkpoint exists.

Definition 1 A *checkpoint*, $C(k_c, k_e)$, is an event that occurs at the checkpoint time, k_c, that first comes into existence at k_e, in which the knowledge set for each robot i contains for all j:

1. the previous centralized-equivalent state estimate of robot j at some timestep, $k_{s,j} \le k_c$,
2. all the odometry and measurement data of robot j from timestep $k_{s,j}$ to k_c.

Equivalently written using mathematics, a checkpoint occurs at timestep k_c when $S_{i,k_e} \supseteq S_{j,k_c}, \forall i \in N, \forall j \in N$.

We want to show that the notion of a checkpoint can be generalized and used in decentralized cooperative SLAM. Suppose that all robots can calculate the centralized-equivalent estimate in decentralized cooperative SLAM at some timestep, k_c. This implies that robots can obtain

$$\begin{aligned}&\mathsf{bel}\left(X_{k_c}\right)\\ &\quad = p\left(X_{k_c}|\mathsf{bel}\left(X_{j,0}\right), \left\{\mathbf{u}_{j,1:k_c}, Y_{j,1:k_c}\right\}, j \in N\right),\end{aligned}$$

or more generally, robots can obtain

$$\begin{cases} p\big(X_{k_c}|\mathsf{bel}(X_{j,k_{s,j}}), \left\{\mathbf{u}_{j,k_{s,j}+1:k_c}\right\}\left\{Y_{j,k_{s,j}+1:k_c}\right\}, j \in N\big), \\ \hfill \text{if } k_{s,j} < k_c. \\ p\left(X_{k_c}|\mathsf{bel}(X_{j,k_{s,j}}), \forall j \in N\right), \text{ if } k_{s,j} = k_c \end{cases}$$

If we examine the dependencies (estimates, odometry, and measurement data) of the above distribution, we can conclude that these dependencies must be in each robot's knowledge set. That is,

$$\begin{aligned} S_{i,k_e} &\supseteq \begin{cases} \{\mathsf{bel}(X_{j,k_{s,j}}), \mathbf{u}_{j,k_{s,j}+1:k_c}, Y_{j,k_{s,j}+1:k_c} | j \in N\}, \\ \hfill \text{if } k_{s,j} < k_c \\ \{\mathsf{bel}(X_{j,k_{s,j}} | j \in N)\}, \text{ if } k_{s,j} = k_c \end{cases} \\ &\supseteq \begin{cases} \{\mathsf{bel}(X_{j,k_{s,j}}), \mathbf{u}_{j,k_{s,j}+1:k_c}, Y_{j,k_{s,j}+1:k_c} | j \in N\}, \\ \hfill \text{if } k_{s,j} < k_c. \\ \{\mathsf{bel}(X_{k_c}) | j \in N\}, \text{ if } k_{s,j} = k_c \end{cases} \end{aligned}$$

The information necessary for all robots to obtain the centralized-equivalent estimate in decentralized cooperative SLAM is the exact same information that is required for a checkpoint to exist. Therefore, we can conclude that the existence of a checkpoint allows all robots to obtain the centralized-equivalent estimate in decentralized cooperative SLAM. More formally, when $C(k_c, k_e)$ exists, all robots can obtain $\mathsf{bel}(X_{k_c})$. Recall that the difference between the checkpoint existence time, k_e, and checkpoint occurrence time, k_c, can be interpreted as the time delay in obtaining the centralized-equivalent estimate. This is a consequence of robots having to operate in a sparsely-communicating network that is never guaranteed to be fully connected.

For detecting the existence of a checkpoint, the following theorem was introduced for the decentralized cooperative localization problem:

Theorem 1.1 *$C(k_c, k_e)$ exists if and only if the knowledge set of each robot at k_e contains $\mathbf{u}_{j,k_c}$ or $\mathsf{bel}(X_{j,k_c})$, $(\forall j)$.*

This theorem also applies the the decentralized cooperative SLAM case. We can show this by proving the theorem again, but it turns out that the proof is exactly the same as in the decentralized cooperative localization problem. Therefore, we have included the proof in the Appendix.

In summary, what we have shown so far is that the concept of a checkpoint applies not only to the decentralized cooperative localization problem, but also to decentralized cooperative SLAM. In other words, we have generalized the notion of a checkpoint, and an outside observer can determine if all robots can obtain the centralized-equivalent estimate in cooperative decentralized SLAM by checking for the existence of a checkpoint. We have also shown that the method for detecting checkpoint existence also extends to the decentralized cooperative SLAM problem. However, the need for a outside observer defeats the purpose of having a decentralized system. Therefore, we need to examine if we can eliminate this need.

4.2 Obtaining the Centralized-Equivalent Estimate—The Local Perspective

Previously, we have looked at how an outside observer can detect when all robots can obtain the centralized-equivalent estimate. We will now look at how a single robot can detect when it can obtain the centralized-equivalent estimate for itself. From [4], we introduced the notion of a partial checkpoint for this purpose.

Definition 2 A *partial checkpoint*, $C_p(k_{c,i}, k_{e,i})$, is an event that occurs for robot i at time $k_{c,i}$, that first comes into existence at $k_{e,i}$, in which the knowledge set for robot i contains for all j:

1. the previous centralized-equivalent state estimate of robot j at some timestep, $k_{s,j} \leq k_{c,i}$,
2. all the odometry and measurement data of robot j from timestep $k_{s,j}$ to $k_{c,i}$.

Equivalently written using mathematics, a partial checkpoint for robot i occurs at timestep $k_{c,i}$ when $S_{i,k_{e,i}} \supseteq S_{j,k_{c,i}}, \; \forall j \in N$.

Similarly to a checkpoint, we want to show that the notion of a partial checkpoint is applicable in decentralized cooperative SLAM. Suppose that robot i can calculate the centralized-equivalent estimate in decentralized cooperative SLAM at some timestep, $k_{c,i}$. This implies that robot i can obtain

$$\begin{aligned}&\mathsf{bel}\left(X_{k_{c,i}}\right)\\&\quad = p\left(X_{k_{c,i}} | \mathsf{bel}\left(X_{j,0}\right), \left\{\mathbf{u}_{j,1:k_{c,i}}, Y_{j,1:k_c}\right\}, j \in N\right),\end{aligned}$$

or more generally, robot i can obtain

$$\begin{cases} p\left(X_{k_{c,i}} | \mathsf{bel}\left(X_{j,k_{s,j}}\right), \left\{\mathbf{u}_{j,k_{s,j}+1:k_{c,i}}\right\} \left\{Y_{j,k_{s,j}+1:k_{c,i}}\right\}, \right. \\ \quad \left. j \in N\right), \quad \text{if } k_{s,j} < k_{c,i}. \\ p\left(X_{k_{c,i}} | \mathsf{bel}\left(X_{j,k_{s,j}}\right), j \in N\right), \; \text{if } k_{s,j} = k_{c,i}. \end{cases}$$

If we examine the dependencies (estimates, odometry, and measurement data) of the above distribution, we can conclude that these dependencies must be in robot i's knowledge set. That is,

$$\begin{aligned}&S_{i,k_{e,i}}\\&\quad \supseteq \begin{cases} \left\{\mathsf{bel}\left(X_{j,k_{s,j}}\right), \mathbf{u}_{j,k_{s,j}+1:k_{c,i}}, Y_{j,k_{s,j}+1:k_{c,i}} | j \in N\right\}, \\ \quad \text{if } k_{s,j} < k_{c,i} \\ \left\{\mathsf{bel}\left(X_{j,k_{s,j}}\right) | j \in N\right\}, \; \text{if } k_{s,j} = k_{c,i} \end{cases}\\&\quad \supseteq \begin{cases} \left\{\mathsf{bel}\left(X_{j,k_{s,j}}\right), \mathbf{u}_{j,k_{s,j}+1:k_{c,i}}, Y_{j,k_{s,j}+1:k_{c,i}} | j \in N\right\}, \\ \quad \text{if } k_{s,j} < k_{c,i}. \\ \left\{\mathsf{bel}\left(X_{j,k_c}\right) | j \in N\right\}, \; \text{if } k_{s,j} = k_c \end{cases}\end{aligned}$$

The information necessary for robot i to obtain the centralized-equivalent estimate in decentralized cooperative SLAM is the exact same information that is required for a partial checkpoint to exist for robot i. In other words, when $C(k_{c,i}, k_{e,i})$ exists for robot i, then robot i can obtain $\mathsf{bel}(X_{k_c})$.

For detecting the existence of a partial checkpoint, we can again use the following theorem that was established for the decentralized cooperative localization problem.

Theorem 2.1 *$C_p(k_{c,i}, k_{e,i})$ exists if and only if the knowledge set of robot i at k_e contains $\mathbf{u}_{j,k_{c,i}}$or $\mathsf{bel}(X_{j,k_{c,i}})$, $(\forall j)$.*

The proof of the above theorem, which indicates that the theorem is applicable to decentralized cooperative SLAM, is in the Appendix.

The following is the key theorem which allows our estimation framework to be decentralized. It states that a robot's decision to invoke the Markov property and discard information (that it has used to generate the centralized-equivalent estimate) does not affect the ability of other robots to perform the same (i.e., it does not affect the existence of checkpoints).

Theorem 2.2 *Suppose $C(k_c, k_e)$ exists, and robot m applies the Markov property when $C_p(k_c, k_{e,m})$ exists (i.e., at $k_{e,m}$). Then $C_p(k_c, k_{e,i})$ continues to exist, $(\forall i)$.*

The proof to this theorem does not change from the decentralized cooperative localization case to the decentralized cooperative SLAM case, and can also be found in the Appendix. The intuition behind this theorem is that robots can communicate their centralized-equivalent estimates to each other instead of passing measurement and odometry data.

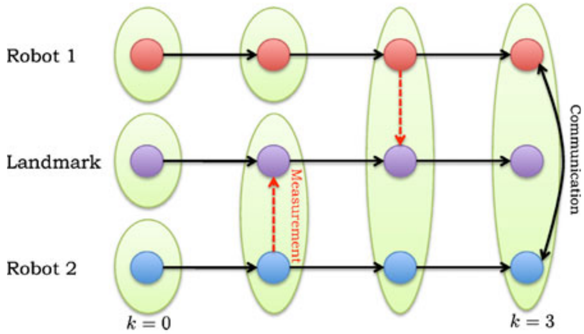
To summarize, a robot can detect whether it can obtain a centralized-equivalent estimate in decentralized cooperative SLAM by checking for the existence of a partial checkpoint, which we have generalized from the decentralized cooperative localization problem. Because of this, we can apply Theorem 2.2 to the decentralized cooperative SLAM problem. The important implication here is that for both the decentralized cooperative localization and decentralized cooperative SLAM problems, a robot can apply the Markov property and discard information that it no longer needs without considering what other robots have in their knowledge sets. In other words, a robot does not need to keep track of the estimates and measurements known to another robot. Yet, we can still guarantee that all robots can obtain the centralized-equivalent estimate. The result of this is that this makes the decentralized cooperative SLAM algorithm similar to the decentralized cooperative localization algorithm. However we need to consider the necessary initial conditions, which differ from decentralized cooperative localization.

4.3 Initial Knowledge of Robots

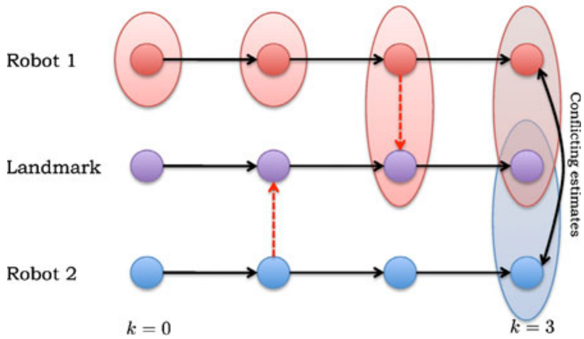
Although we have extended the applicability of the theorems developed for decentralized cooperative localization to decentralized cooperative SLAM, these two problems do not share the same requirements on the initial knowledge of robots. In decentralized cooperative localization, when we have the communication range limit greater than the observation (measurement) range limit, it is unnecessary for a robot to know the total number of robots in the team and yet it can still obtain the centralized-equivalent estimate. This is because the estimates for robot subgroups that have not observed each other are statistically independent. This allows us to combine the estimates from two groups, $\mathsf{bel}(X_{Q_1,k})$ and $\mathsf{bel}(X_{Q_2,k})$, as follows [23]:

$$\mathsf{bel}\left(X_{Q_1,k}, X_{Q_2,k}\right) = \mathsf{bel}\left(X_{Q_1,k}\right)\mathsf{bel}\left(X_{Q_2,k}\right) \qquad (3)$$

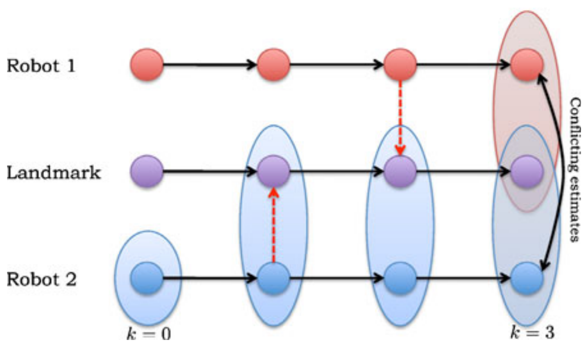
Obtaining the centralized-equivalent estimate in decentralized cooperative SLAM is more restrictive, as the number of robots in the system needs to be known to all robots. To motivate this idea, suppose that we separate the robots into subgroups that initially do not know of each other's existence for the decentralized cooperative SLAM problem. In this case, we are not guaranteed that the subgroup estimates are statistically independent due to the presence of landmarks. Consider Fig. 2 as an example, where there are two sub-groups consisting of a single robot each.



(a) Correlations in the centralized estimates.



(b) Correlations in robot 1's decentralized estimates.



(c) Correlations in robot 2's decentralized estimates.

Fig. 2 In the above information flow graphs with measurements explicitly labelled, the correlations between the estimates are shown as shaded ellipses for: **a** the centralized estimator, which have the estimates correctly correlated, **b** the decentralized estimates made by robot 1 assuming that it initially only knows of its own existence, and **c** the decentralized estimates made by robot 2 assuming that it initially only knows of its own existence. In the latter two cases, the robots will have applied the Markov property at every timestep. When the two robots finally communicate with each other, they will possess decentralized estimates that cannot be used to recover the centralized-equivalent estimate

Acting independently, each robot's self-estimate will be correlated with the landmark, but they will not be aware that the landmark has been observed by both robots (i.e., the estimate of both robots and the landmark should all be correlated) until they encounter each other. Each robot will also apply the Markov property at every timestep because they only know of their own existence. When the two robots finally encounter each other, they will not be able to merge their estimates to form the centralized-equivalent estimate. To formalize the above idea, we have the following theorem:

Theorem 3.1 *In decentralized cooperative SLAM, assume robot i does not initially know the total number of robots in the team. If robot i detects a partial checkpoint, $C_p(k_{c,i}, k_{e,i})$, based on robots known to it, $N_{i,k_{e,i}}$, and applies the Markov Property when the partial checkpoint exists, then robot i is not guaranteed to obtain the centralized-equivalent estimate for all robots and known landmarks when it knows of the existence of all robots.*

Proof Since we want to show that the centralized-equivalent estimate is not guaranteed to be obtainable by robot i, we only need to show one example of when this is the case. We know that robot i knows of its own existence, and therefore, $i \in N_i$. Now suppose robot i takes a measurement, $\mathbf{y}^{m,i}_{i,k_m}$, of a landmark, m. Suppose that another robot, $n \in \overline{N_i}$ (not known to robot i), also takes a measurement, $\mathbf{y}^{m,n}_{n,k_m}$, of the same landmark. We can show that robot i will know of landmark m when it detects the existence of the partial checkpoint $C_p(k_m, k_{e,i})$:

$$\begin{aligned}\mathbf{y}^{m,i}_{i,k_m} \text{ exists} &\Rightarrow \mathbf{y}^{m,i}_{i,k_m} \in S^-_{i,k_m}\\ &\Rightarrow \mathbf{y}^{m,i}_{i,k_m} \in S_{i,k_m}\\ &\Rightarrow \mathbf{y}^{m,i}_{i,k_m} \in S_{i,k_{e,i}}\\ &\Rightarrow m \in M_{i,k_{e,i}}\end{aligned}$$

Robot i however, will not know that robot n exists or that it has also made a measurement of landmark m, unless robots i and n were within the communication range when the measurement was made of landmark m at timestep k_m. This is not guaranteed to happen because we do not enforce that robots must remain within communication range of each other. Mathematically, suppose $\mathbf{y}^{m,n}_{n,k_m}$ exists, and that $d^{i,n}_k > r_{\mathsf{comm}}$:

$$\begin{aligned}d^{i,n}_k > r_{\mathsf{comm}} &\Rightarrow \mathbf{y}^{m,n}_{n,k_m} \in S_{n,k_m} \text{and } \mathbf{y}^{m,n}_{n,k_m} \notin S_{i,k_{e,i}}\\ &\Rightarrow n \notin N_{i,k_{e,i}}\end{aligned}$$

Now if robot i detects partial checkpoints based on robots known to it and applies the Markov property when a partial checkpoint exist, then its estimate, $\mathsf{bel}(X_{i,k_m})$, cannot be used to calculate the centralized-equivalent-estimate for all robots and known landmarks using Eq. 3 because of the correlation (i.e., the measurement made by robots i and n of landmark m) of which robot i does not know. In other words,

$$\begin{aligned}&\mathsf{bel}\left(X_{i,k_m}, \overline{X_{i,k_m}}\right) \neq \mathsf{bel}\left(X_{i,k_m}\right) \mathsf{bel}\left(\overline{X_{i,k_m}}\right)\\ &\Rightarrow \mathsf{bel}\left(X_{i,k}, \overline{X_{i,k}}\right) \neq \mathsf{bel}\left(X_{i,k}\right) \mathsf{bel}\left(\overline{X_{i,k}}\right), k \geq k_m\end{aligned}$$

Note here that we cannot calculate the centralized-equivalent estimate from the original measurement and odometry data since they have been discarded when the Markov property was applied. Therefore, not only is robot i unable to use $\mathsf{bel}(X_{i,k_m})$ to determine the centralized-equivalent estimate for all robots and known landmarks at timestep k_m, future estimates calculated by recursive filtering also cannot be used to recover the centralized-equivalent estimate for all robots and known landmarks. □

Corollary 3.1 *In decentralized cooperative SLAM, all robots must initially know the total number of robots in the team to guarantee that all robots can obtain the centralized-equivalent estimate for all robots and known landmarks in decentralized cooperative SLAM.*

Proof From Theorem 3.1, we know that a robot is not guaranteed to obtain the centralized-equivalent estimate for all robots and known landmarks if the number of robots in the team is initially unknown, and if it detects partial checkpoints based on the number of robots known to it. Even if only one robot in the team does not initially know the total number of robots, there is the possibility that its estimate (which cannot be used to recover the centralized-equivalent estimate for all robots and known landmarks) will be passed to other robots. On the other hand, suppose that all robots initially know the total number of robots in the team, and only use the existence of partial checkpoints (based on having the required information from all robots) to determine when the centralized-equivalent estimate is obtainable (and when the Markov property can be applied). In this case, all robots are guaranteed to obtain the centralized-equivalent estimate for all robots and known landmarks. □

In summary, in decentralized cooperative SLAM we cannot make use of the notion of independent subgroups, and we cannot guarantee that a robot can obtain the centralized-equivalent estimate (over all robots and known landmarks) without detecting a partial checkpoint based on information from all robots, which requires the robot to know the number of robots in the network. This is not a requirement in decentralized cooperative localization, but is necessary for decentralized cooperative SLAM due to the presence of landmarks. Although this may seem restrictive, we must remember that this is because we want all robots to be able to obtain the centralized-equivalent estimate in a sparsely-communicating robot network that is never guaranteed to be fully-connected.

An alternative way of ensuring that the centralized-equivalent estimate is obtainable is to keep all robots from applying the Markov property (i.e., have all robots retain all information in their knowledge set through time). This, however, is impractical as computation and memory usage will increase indefinitely. Hence, we need to allow robots to know when they can apply the Markov property, which requires knowing the number of robots in the system.

Overall, we have generalized the concepts of a checkpoint and a partial checkpoint, allowing them to be applicable to the decentralized cooperative SLAM problem. The existence of a checkpoint is equivalent to when it is possible for all robots to obtain the centralized-equivalent estimate at the checkpoint occurrence time. Similarly, the existence of a partial checkpoint for a particular robot is equivalent to when it is possible for that robot to obtain the centralized-equivalent estimate. From this, we are then able to reuse theorems from [4], which allow robots to apply the Markov property without consideration of other robots, while still ensuring that all other robots can obtain the centralized-equivalent estimate. However, we require robots to initially know the total number of robots in the system.

5 Decentralized Cooperative SLAM Algorithm

Algorithm 1 is our proposed decentralized cooperative SLAM algorithm, developed based on the theory provided in Section 4. Figure 3 is a graphical representation of the algorithm.

Algorithm 1 Decentralized cooperative SLAM(k, $\mathbf{u}_{i,k}$, $Y_{i,k}$, $S_{i,k-1}$, $S_{j,k}$ $(i \in N)(\forall j)(d_k^{j,i} \leq r_{\mathsf{comm}})$)

1 $S_{i,k} \leftarrow S_{i,k-1} \cup \{\mathbf{u}_{i,k}\} \cup \{Y_{i,k}\} \bigcup_{j \in R_{i,k}} S^{-}_{j,k}$;
2 $K \leftarrow \{k_r\}$such that$\{\mathbf{u}_{i,k_r}\} \in S_{i,k} (\forall i \in N)$;
3 $k_{c,i} \leftarrow \mathsf{max}(K)$;
4 $\tilde{S}_{i,k_{c,i}} \leftarrow S_{i,k} - \left\{\mathbf{u}_{j,k_r}, Y_{j,k_r}\right\}$ $(\forall j, \forall k_r > k_{c,i})$;
5 $\mathsf{bel}^* \left(X_{k_{c,i}}\right) \leftarrow p\left(X_{k_{c,i}} | \tilde{S}_{i,k_{c,i}}\right)$;
6 $S_{i,k} \leftarrow S_{i,k} \cup \mathsf{bel}^* \left(X_{k_{c,i}}\right)$;
7 $S_{i,k} \leftarrow S_{i,k} - \left\{\mathbf{u}_{j,k_r}, Y_{j,k_r}, \mathsf{bel}^* \left(X_{k_r}\right)\right\} (\forall k_r \leq k_c, \forall j)$;
8 $\mathsf{bel} \left(X_k\right) \leftarrow p\left(X_k | S_{i,k}\right)$;
9 **return** $\left\{\mathsf{bel} \left(X_k\right), S_{i,k}\right\}$;

Algorithm 1 executes at every timestep on all robots. On line 1, we update the knowledge set of robot i by implementing Eqs. 1 and 2. In lines 2 and 3, we search for the latest partial checkpoint. Line 4 defines the subset of knowledge that includes all information up to the partial checkpoint time. On line 5, we calculate an estimate for the partial checkpoint time. Note that bel^* indicates the centralized-equivalent estimate. On line 6, we proceed to discard information replaceable by bel^* (invoking the Markov property), and enter the newly-calculated belief into

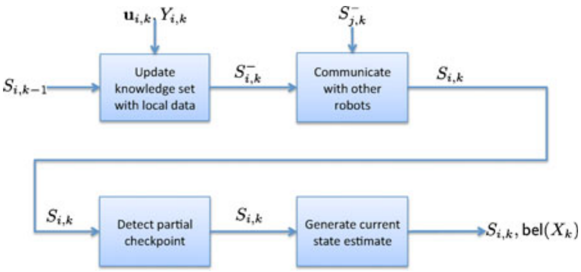


Fig. 3 The decentralized cooperative SLAM algorithm, which provides centralized-equivalent state estimates of all robots and map features whenever possible

the knowledge set on line 7. On line 8, we use all available information in the knowledge set to produce the current state estimate. Note that because the centralized-equivalent estimate can be delayed, we produce a temporary estimate until the centralized-equivalent estimate can be obtained. For the temporary estimate, robot i assumes robot j maintains its last known velocity if odometry data is not available. Due to this assumption, temporary estimates for robots that have been out of communication range for an extended period of time may have diverged from the real robot pose. Nevertheless, we are guaranteed that the centralized-equivalent estimate can be recovered when communication is reestablished.

Aside from being able to generate the centralized-equivalent estimate in a network that is never guaranteed to be fully connected, another feature of our cooperative decentralized SLAM algorithm is that we can use a variety of recursive filtering and data association methods within our framework (on lines 5 and 8). The framework is not dependent on specific filtering methods such as the EKF, but the underlying assumption is that the Markov property is valid and applicable (i.e., any recursive approximation of the Bayes filter could be used within our framework).

5.1 Complexity Analysis

Complexity analysis of our decentralized cooperative SLAM algorithm is difficult because it is not possible to predict the connectivity of a dynamic robot network. Overall, the complexity of the algorithm is at least on the order of the filtering and data association methods implemented, and the computational requirement for a robot will increase as the number of timesteps since its last partial checkpoint (i.e., $k - k_c$) increases. Let us consider the worst-case scenario as shown in Fig. 4 where we have n robots and m landmarks. In this scenario, we have every robot measuring every other robot and all landmarks at every timestep. However, one robot does not communicate with the rest of the team. We will assume that the *Extended Kalman Filter (EKF)* [23] is used as the filtering method. The state of the system has a dimension that is proportional to the number of robots, n, and the number landmarks, m. At each timestep the number of measurements is on the order of $n^2 + nm$, and the computational complexity for processing each measurement is quadratic with respect to the number of measurements. This implies that computational complexity is $O(n^4 + n^3m)$ for the centralized estimator in the worst-case scenario. Furthermore, storing the covariance matrix and measurements requires memory usage of $O\left((n+m)^2\right)$.

For the same worst-case scenario, the complexity for calculating the current estimate with our decentralized cooperative SLAM algorithm is $O\left((k - k_c)(n^4 + n^3m)\right)$ when a new partial checkpoint is discovered. The $k - k_c$ (delay) factor comes from the fact that we need to run the recursive filter from time k_c to the current time k.

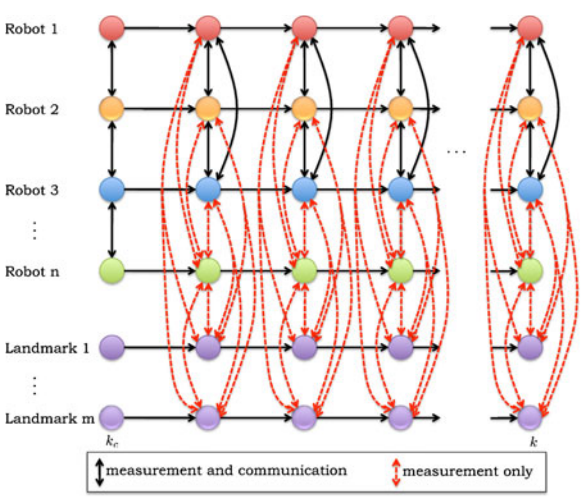


Fig. 4 The worst-case scenario in decentralized cooperative SLAM in terms of memory usage and computational complexity

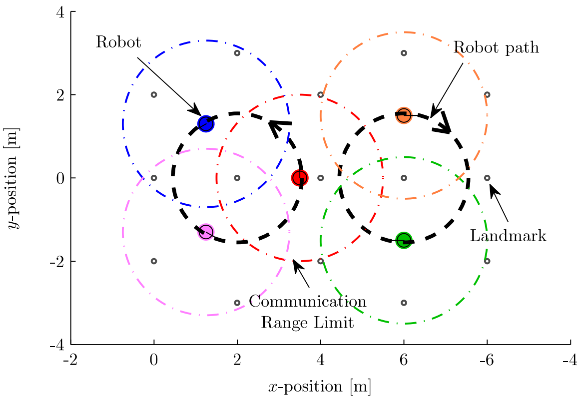
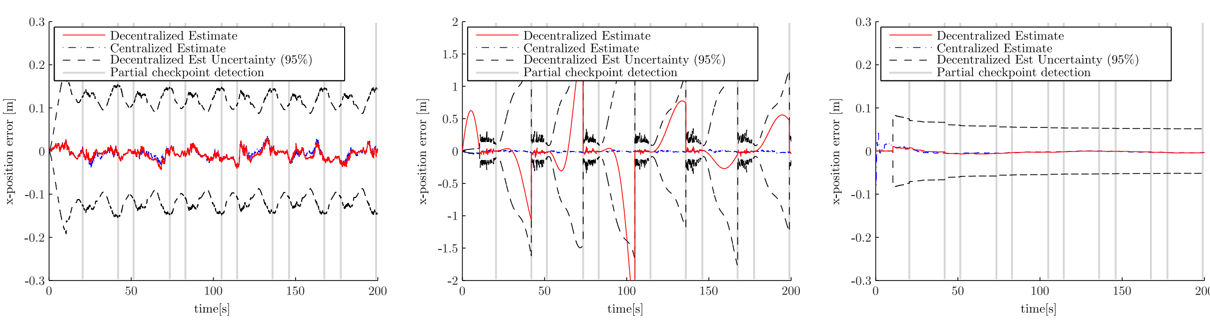


Fig. 5 A simulation of a weaving robot network that is never fully connected

For the case where no new partial checkpoint is discovered, and by knowing the state estimate from the previous timestep $(k-1)$, computational complexity is $O(n^4+n^3m)$ (i.e., the same as the centralized estimator). The memory usage requirement for the decentralized estimator is $O\left((k-k_c)(n+m)^2\right)$. Again, the $k-k_c$ factor comes from the fact that we need to keep all the information since the previous partial checkpoint time to ensure that the centralized-equivalent estimate is obtainable. Lastly, the communication bandwidth requirement is $O((k-k_c)(n^3+n^2m))$ for our decentralized approach. In practice, we can reduce communication bandwidth by having each robot remember the latest information sent to other robots (but note that this does not imply having to know what each robot has in their knowledge set).

5.2 Simulation

We will use a simulation as a first validation of our decentralized cooperative SLAM algorithm. In the scenario, illustrated in Fig. 5, the robots are moving in circles at constant forward and angular velocities, and the communication range is limited such that a 'weaving' pattern emerges from the communication linkages established by the robots in both circles. This also prevents the network from ever being fully connected. As a consequence, there will be a delay to when the centralized-equivalent estimate for the current timestep can be obtained (i.e., we need to wait until a partial checkpoint occurs for the current timestep). This also implies that the estimates shown are the temporary estimates made at each timestep with the information available to the robot. Recall that we assume other robots maintain their last known forward and angular velocities if this information is not available. Nevertheless, our algorithm guarantees that the centralized-equivalent estimate can always be recovered. Figure 6 shows several components of the (current, and temporary) decentralized estimate from robot 1 (red). Note that even though there is a delay to when the centralized-equivalent estimate is obtainable, its (current and temporary) estimate of its own x-position as well as a landmark's x-position is very close to the centralized estimate.



(a) Robot 1's estimate of its own x-position. (b) Robot 1's estimate of robot 3's x-position. (c) Robot 1's estimate of a land -mark's x-position.

Fig. 6 Components of robot 1's decentralized estimate for the scenario shown in Fig. 5

Robot 1's estimate of another robot, however, exhibits greater differences due to the loss of communication. Although this simulation scenario is simple, it nevertheless demonstrates that our algorithm can be used in a network that is never fully connected. Next, we will discuss an implementation our decentralized cooperative SLAM algorithm on real hardware.

6 Experimental Setup

Our decentralized approach to SLAM was validated through real hardware experiments. Figure 7 shows our equipment, which includes a fleet of robots and a collection of landmarks. A 10-camera Vicon motion capture system is used to provide groundtruth data for both robots and landmarks at 100Hz with millimeter accuracy [24].

All five robots in the fleet are identical in construction, and built from the iRobot Create (two-wheel differential drive) platform. Each robot is equipped with a laptop computer, which runs the Player [25] middleware to perform motion control, image processing, and data logging. A monocular camera serves as the primary sensing instrument on each robot. Camera images are processed by a custom Player driver to identify cylindrical barcode tubes on all robots and landmarks for data association, and to obtain range and bearing measurements.

The experiments were carried out in an area of approximately 15 m × 8 m. The five robots and 15 landmarks were placed inside the workspace in a variety of configurations. During the experiments, the robots drove randomly in the lab space while avoiding obstacles and each other. At the same time, the robots recorded all their odometry data, as well as range and bearing measurements with timestamps. The clocks on the robots were synchronized by running the *Network Time Protocol daemon (NTPD)* on each robot. From the experimental trials, nine sub-datasets were collected, which are publicly available (please refer to [26] for detailed information). The duration of each sub-dataset range from approximately 20–70 min, and we have collected and processed 250 min of data. Some additional test trials were used to calibrate our motion model and measurement model and to characterize their noise properties.

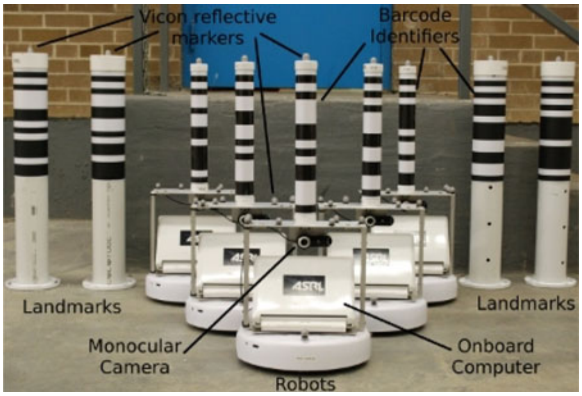


Fig. 7 Robots and landmarks for our decentralized cooperative SLAM experiments

7 Experimental Results

Communication range is an important factor to consider when we interpret the results from our decentralized estimator. As communication range increases, we can expect the percentage of time that the network is fully connected to also increase. Furthermore, we can expect that robots have more opportunities to communicate with each other. For each of our sub-datasets, the percentage of time that the network is fully connected is shown in Fig. 8. The zero-to-one transition phenomenon observed in the figure is typical for a *fixed-radius-random-graph-model* [27], which we are using for generating our robot network. We have indicated on the figure the data that correspond to sub-dataset 1 of our experiments, as we will be examining the performance of our decentralized estimator for this test in detail. We picked this test in particular because it displayed the lowest percentage of full connectivity time for all communication range limits out of all the sub-datasets (i.e., sub-dataset 1 is the worst-case).

For the following results, the observation range, r_{obs}, is 3m, while communication interval is limited to 0.5 s. The EKF is used as the filtering method in our decentralized cooperative SLAM algorithm, which runs on all robots for estimating the pose (x, y, θ) and position (x, y) of all robots and landmarks, respectively. The performance of

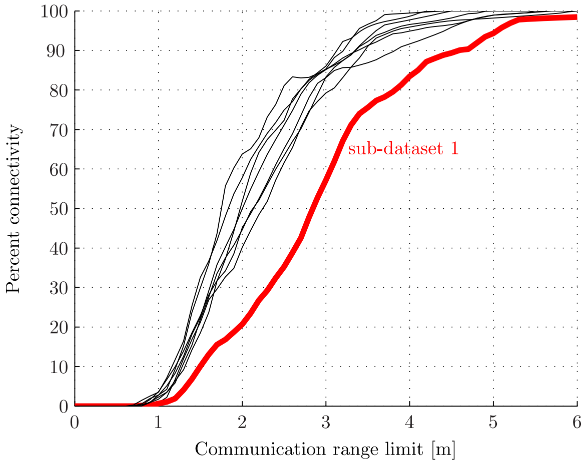


Fig. 8 Percentage of time when the network is fully connected as a function of communication range limit in each experimental trial

our algorithm is compared against that of the centralized EKF-SLAM algorithm. In order for the centralized estimator to work, we allow it to cheat by ignoring the communication constraints (i.e., the robot network is connected at all times for the centralized estimator). Since this paper is focused on the state estimation aspect of decentralized cooperative SLAM, we performed data association using barcode identification. Note, however, that our framework allows for other data association techniques to be implemented [28].

We will begin by examining the results from test 1 in detail. In particular, we will look at robot 1's estimates. Detailed results from the other robots are similar and are not shown. Figure 9 shows the results of our decentralized cooperative SLAM algorithm at 30 min into sub-dataset 1. In this case, the communication range was limited to 2 m, and the network was fully connected only 21% of the time.

Figure 10 displays the memory usage of robot 1 in running our algorithm on sub-dataset 1, for the cases where the communication range limit is limited to 1, 2, and 3 m. Referring to Fig. 8, these cases correspond to 0.6, 20.7, and 52.7% of full network connectivity, respectively. Figure 10 also shows the hypothetical memory usage of robot 1 had it never applied the Markov property. In using our decentralized cooperative SLAM algorithm, we are able to limit memory usage because our algorithm makes use of the Markov property. Temporary increases in memory usage are caused by the loss of connectivity to other robots in the network (i.e., robot 1 needs to retain information since its last partial checkpoint occurrence time). These increases are followed by sharp decreases, which signal that robot 1 has applied the Markov property (i.e., robot 1 has detected a new partial checkpoint). Note, however, that decreases in memory use do not necessarily imply that the network is fully connected. For the case where $r_{\mathrm{comm}} = 1$ m, the network is only fully connected 0.6% of the time. Yet, robot 1 is still able to occasionally apply the Markov property because information from other robots can be relayed. As the communication range limit increases, we can see that memory usage decreases, and the

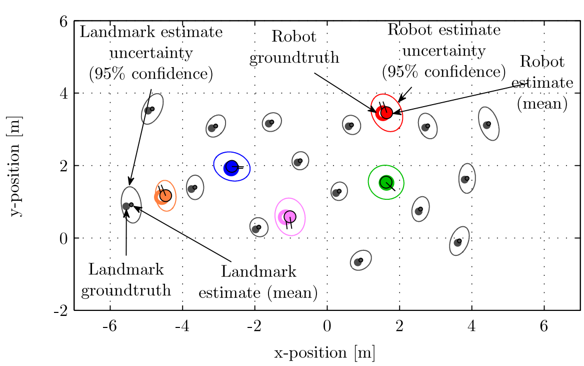


Fig. 9 A graphical representation of the results from test 1 after 1,800 [s]

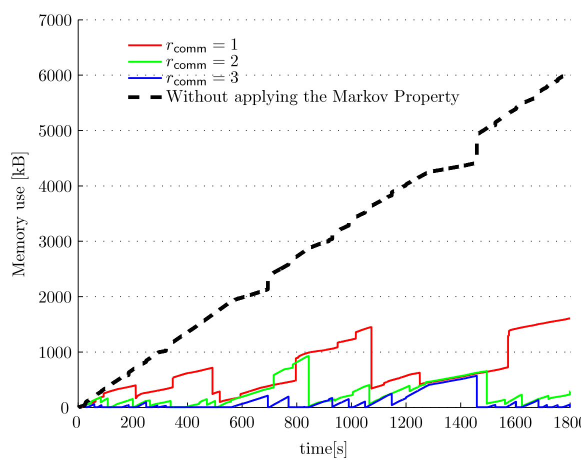


Fig. 10 Memory usage of robot 1 in test 1

Markov property is applied more frequently. This is because robots have more opportunities to communicate with each other as the communication range limit increases. Similar trends in memory usage are observed for all other robots, and in all other tests.

Figure 11 shows the amount of information communicated with robot 1 in test 1 for cases with different communication range limits. The amount of information communicated remains relatively low for most timesteps in all cases. Instances where there is a significant increase in the amount of information communicated indicate that robot 1 has just re-established connection with one or more robots (i.e., the robots are updating each other with the latest information in their knowledge sets). As the communication range limit decreases, robots are more likely to lose connection with other robots and experience longer durations of disconnection. This explains why more information is exchanged when communication is re-established.

Figure 12 contains a collection of graphs showing several components of robot 1's estimate error for the cases where the communication range limit is limited to 1, 2, and 3 m. The components shown include robot 1's estimate error of its own x-position, robot 1's estimate error of robot 3's x-position, and robot 1's estimate error of a landmark's x-position. These components shown are representative of a robot's estimate error of its own pose (for which odometry information

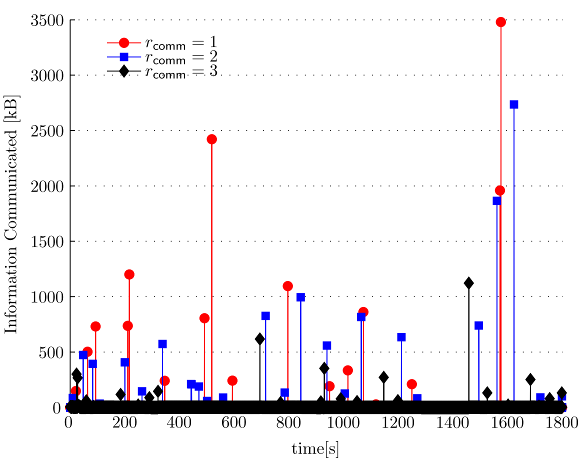
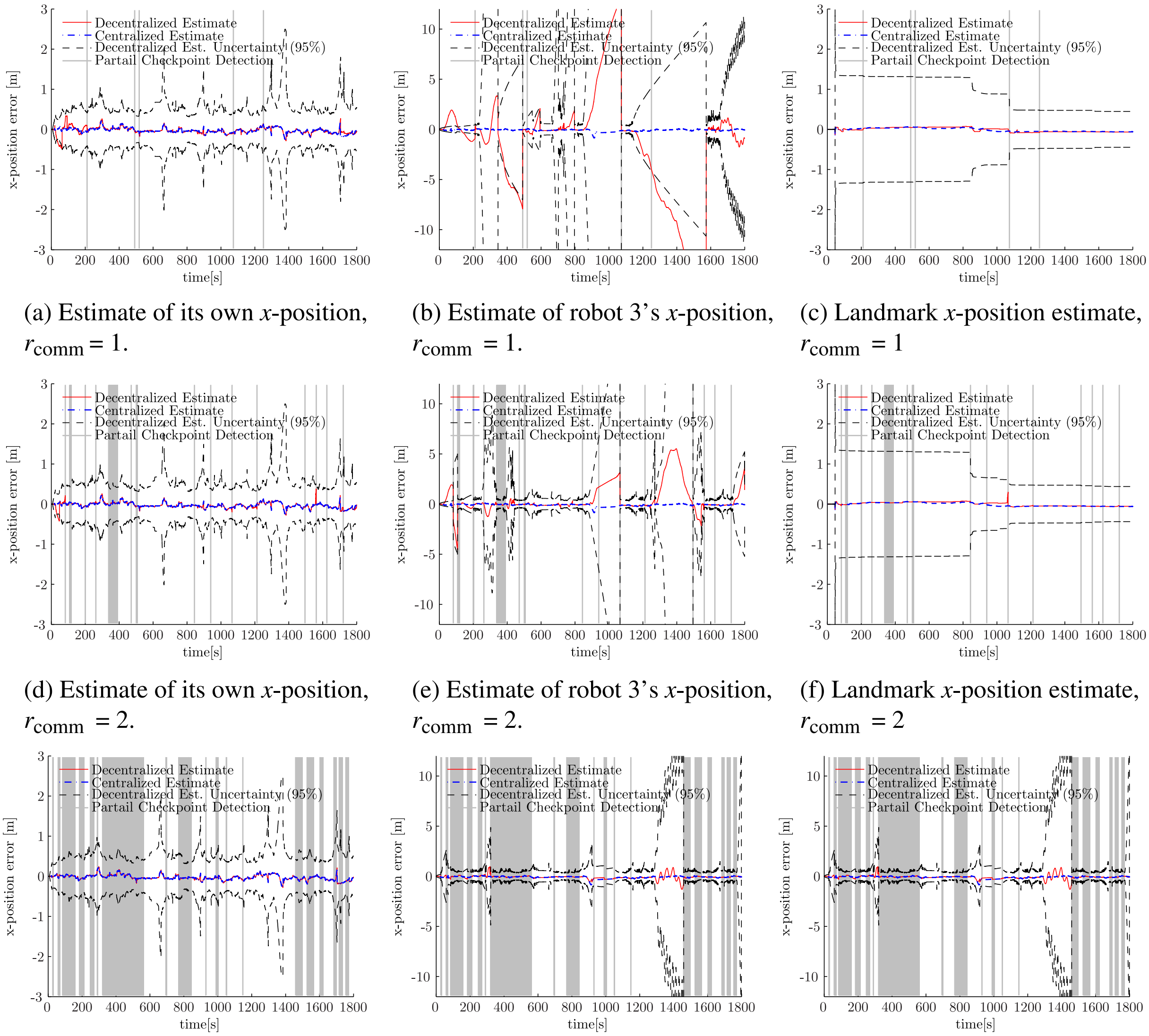


Fig. 11 Information exchanged with robot 1 in test 1

is always available), a robot's estimate error of another robot's pose (for which odometry data are not always available), and a robot's estimate error of a landmark position. In each of the plots, we also compare with the centralized estimator error. It is important to remember that we are allowing the centralized estimator to cheat by ignoring the communication restrictions. The uncertainty regions shown on each plot are projections of the 95% confidence ellipsoid for the full-state decentralized estimate (i.e., including all robots and landmarks). The grey areas in the plots indicate the timesteps at which partial checkpoints were detected. In other words, robot 1 is able to obtain centralized-equivalent estimates at these timesteps. However, note that what is shown is the existence of partial checkpoints at the indicated timesteps. The occurrence time of these partial checkpoints may be delayed depending on the network connectivity. Hence, even if partial checkpoint existence is detected, the estimate shown may still be a temporary estimate.

In Fig. 12a, d, g, and c, f, i, we can see that robot 1's self-estimate and its estimate of a landmark match closely with the centralized state estimates, even in the $r_{\mathsf{comm}} = 1$ m communication range limit case where the network is only fully connected 0.6% of the time. Larger differences can be observed between the centralized and decentralized for robot 1's estimate of robot 3's x-position in Fig. 12b, e, h. Instances of larger differences occur due to the loss of communication between robot 1 and 3. During this time robot, 1 assumes the last known velocity of robot 3 to calculate an estimate. However, this estimate is only temporary. The difference between the decentralized and centralized estimate decreases eventually, when robot 1 communicates with robot 3 again, or obtains information of robot 3 through another robot. As the communication range limit increases, the duration and frequency of communication loss between robot 1 and robot 3 decrease. This leads to smaller differences observed between the decentralized and centralized estimates. Estimation errors that are similar to the above figures are observed in the state estimates of the other robots. Note here that we cannot expect the decentralized and centralized estimates to be the same unless a partial checkpoint occurs



(a) Estimate of its own x-position, $r_{\text{comm}} = 1$.
(b) Estimate of robot 3's x-position, $r_{\text{comm}} = 1$.
(c) Landmark x-position estimate, $r_{\text{comm}} = 1$
(d) Estimate of its own x-position, $r_{\text{comm}} = 2$.
(e) Estimate of robot 3's x-position, $r_{\text{comm}} = 2$.
(f) Landmark x-position estimate, $r_{\text{comm}} = 2$
(g) Estimate of its own x-position, $r_{\text{comm}} = 3$.
(h) Estimate of robot 3's x-position, $r_{\text{comm}} = 3$.
(i) Landmark x-position estimate, $r_{\text{comm}} = 3$

Fig. 12 Components of robot 1's decentralized estimates at various communication range limits

at the latest timestep (i.e., if a robot communicates with all other robots at the current timestep). In all other cases, the partial checkpoint occurrence time (i.e., the time of the centralized-equivalent estimate) is delayed, and we need to use a temporary estimate for the current timestep.

We will now turn our attention to looking at the overall results from all the datasets. Again, we tested our decentralized cooperative SLAM algorithm using communication range limits of 1m, 2m, and 3m. Each data point in Fig. 13 corresponds to either the average memory use by all robots (red circles) or the maximum memory use out of all robots (blue squares) in a particular test, under a particular communication range limit. A line of best fit is also plotted for both average memory use and maximum memory use to illustrate the general trend of the data. We can observe that as communication range decreases (or as the percentage of time that the network is fully

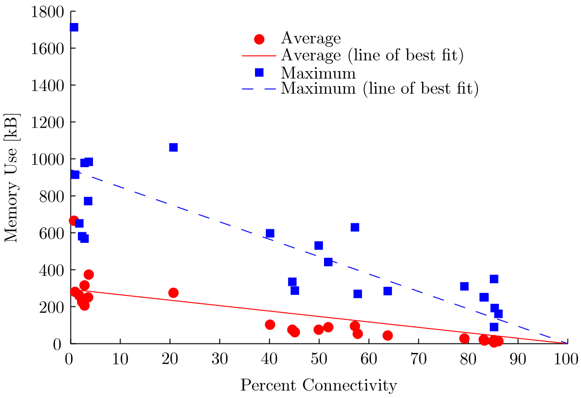


Fig. 13 Average and maximum memory usage as a function of the percentage of time that the network is fully connected

connected increases), both average and maximum memory use decrease. This corresponds with the trend observed in Fig. 10. To reiterate, this trend is due to the robots applying the Markov property more frequently as communication range limit increases.

Figure 14 shows the average amount of information communicated between robots during each communication interval (every 0.5 s), as well as the maximum amount of information communicated out of all robots. Again, each data point corresponds to a particular test and is restricted to a particular communication range limit. The average amount of information communicated remains low regardless of the percentage of time that the network is fully connected. The maximum

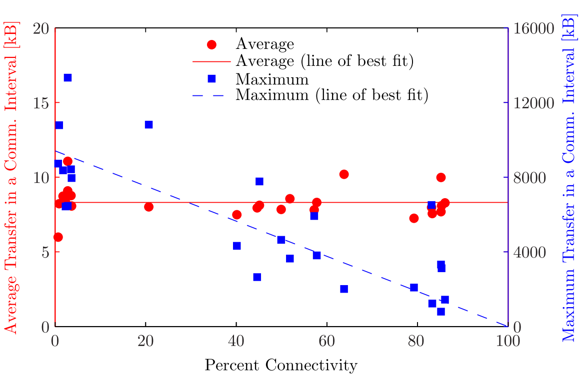


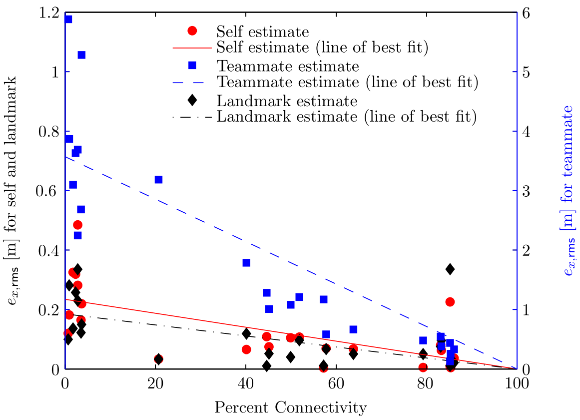
Fig. 14 Average and maximum amount of information communicate between robots as a function of the percentage of time that the network is fully connected

amount of information communicated increases as the percentage of time that the network is fully connected decreases. The reason for this, as explained previously, is because robots are likely to lose connection with other robots and experience longer duration of disconnection as the communication range limit decreases. These observations are consistent with the results in Fig. 11. Although Fig. 14 shows that the amount of information communicated can become high when the communication range limit is low, this is only the case if robots have to update each other with the newest information within one communication interval. In reality, robots are usually within communication range of each other over multiple communication intervals. Hence, information exchange could be spread out over this duration.

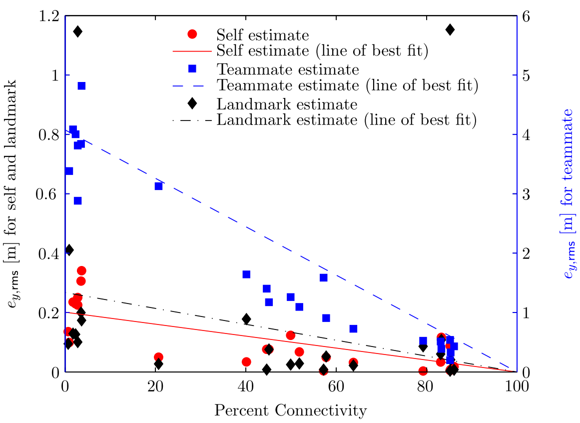
Next, we will look at the differences between the decentralized and centralized estimates in all eight tests. Figure 15a–c show the root-mean-square (rms) differences between the decentralized and centralized x-position estimates, $e_{x,\mathrm{rms}}$, y-position estimates, $e_{y,\mathrm{rms}}$, and θ-orientation estimates, $e_{\theta,\mathrm{rms}}$, respectively. In each plot, we also distinguish between robots' estimates of themselves (red circle), robots' estimates of all other teammates (blue square), and robots' estimates of all landmarks (black diamond). Lines of best fit are plotted to provide a sense of the general trends in the data. These trends are consistent with what was observed in the detailed results from test 1 presented earlier.

In general, for a robot's decentralized estimates of its own pose, there is very little difference compared to the centralized estimates. Even as communication range limit decreases, the differences between the decentralized and centralized estimates only increase slightly. This is because a robot's own odometry and measurements are always known. The increasing differences between the decentralized and centralized self-estimates with decreasing percent connectivity can be explained by the reduced chance for a robot to obtain measurements from other robots as the communication range limit decreases.

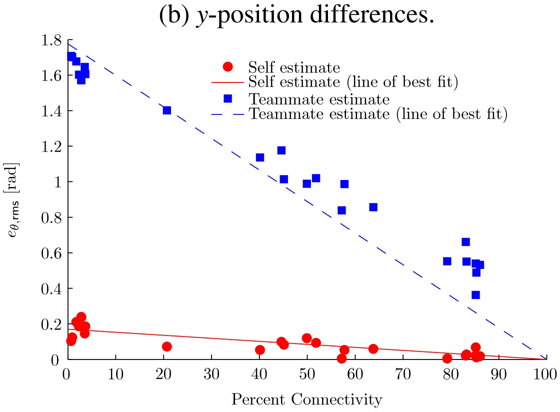
For a robot's estimates of other robots' poses, there is on average a greater difference between the centralized and decentralized estimate compared to the error in its self-estimate. This is



(a) x-position differences.



(b) y-position differences.



(c) θ, orientation, differences.

Fig. 15 The rms differences between the decentralized and centralized estimate for a robot's self-estimate, estimates of teammates, and estimates of landmarks, as a function of the percentage of time that the network is fully connected

because the latest odometry data from other robots are not always available for a robot's current state estimate. Furthermore, this difference becomes larger as the communication range limit decreases since the opportunities for communication between robots also decrease. Note, however, that the current state estimate is temporary and can be updated when more information becomes available. Our decentralized cooperative SLAM algorithm ensures that the centralized-equivalent estimate can be obtained later (when a partial checkpoint is detected).

For landmarks, the average differences between the decentralized and centralized estimates follow a similar trend to the robots' self-estimates. The differences are relatively small because landmarks are static (i.e., they do not have odometry data), and are mainly due to a robot not having the latest measurements from other robots. This is an important point, as it indicates that a robot can retain a good estimate of its own state and the location of all landmarks even when network connectivity is sparse. This is useful to know should robots need to perform path planning in the workspace.

8 Conclusions

In this paper, we posed the centralized-equivalent decentralized cooperative SLAM problem, where it is necessary for all the robots in a team to estimate the position of the landmarks in a workspace, as well as the pose of all robots, in a network where full connectivity is never guaranteed. The novel contributions of this paper include:

1. A decentralized cooperative SLAM algorithm that allows each robot to obtain the exact centralized-equivalent estimate for all robots and known landmarks whenever possible, even if the network is never fully connected. This algorithm also ensures that a robot only needs to consider its own knowledge before applying the Markov Property to discard information that it no longer needs, while ensuring that all other robots can still obtain the centralized-equivalent estimate. Hence, a robot does not need to keep track of what other robots know.

2. Extensive hardware experiments, the results of which are used to validate our approach. We have performed over 250 min of hardware experiments using a fleet of five robots where we have accurate groundtruth data of all robot poses and all landmark positions (which is rarely available for SLAM experiments).
3. An analysis of how network connectivity affects estimation performance, and a demonstration that a robot's decentralized estimate of the map and its own pose always remains close to the centralized-equivalent estimate even when robots rarely communicate with each other.

In developing our algorithm, we generalized the concepts of a checkpoint and a partial checkpoint, making them applicable to both the decentralized cooperative localization problem, and the decentralized cooperative SLAM problem, which we examined in this paper. These events identify when all robots and when a single robot can obtain the centralized-equivalent estimate of the system, respectively. The generalization is important as it allows us to use theorems previously developed for the decentralized cooperative localization problem. Furthermore, we proved that it is necessary in decentralized cooperative SLAM for each robot to initially know the total number of robots in the network. This initial condition is not necessary for the cooperative localization case but is required in decentralized cooperative SLAM. Although this may seem somewhat restrictive, it is necessary for ensuring that the centralized-equivalent estimate is obtainable. Our centralized-equivalent approach allows proper handling of out-of-sequence measurements and avoid cyclic update problems.

With our extensive hardware experiment, we were able to compare the results from our algorithm against those from a centralized estimator (which we allowed to cheat by ignoring communication constraints), and tested our algorithm under different communication range limits. Our results show how memory usage is limited in our algorithm due to use of the Markov property. The results also show that the centralized-equivalent estimate can always be recovered after a period of poor network connectivity. In terms of accuracy, our overall results show that a robot's estimate of its own pose and the position of landmarks are close to the centralized estimate even with a low percentage of time in which the network is fully connected. The accuracy of a robot's estimate of another robot depends on the percentage of time in which the network is fully connected.

Acknowledgement This work received supports from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Appendix: Proofs

Proof of Theorem 1.1 Expressed mathematically the theorem statement is:

$$\begin{aligned} & S_{i,k_e} \supseteq S_{j,k_c}, \quad (\forall i, j) \\ \Leftrightarrow\ & S_{i,k_e} \supseteq (\mathbf{u}_{j,k_c} \text{ or } \mathsf{bel}(X_{j,k_c})), \quad (\forall i, \forall j) \end{aligned}$$

We will approach this proof by first assuming that $C(k_c, k_e)$ exists:

$$\begin{aligned} & S_{i,k_e} \supseteq S_{j,k_c}, \quad (\forall i, \forall j) \\ \Rightarrow\ & S_{i,k_e} \supseteq \begin{cases} \left\{\mathsf{bel}(X_{j,k_{s,j}}), Y_{j,k_{s,j}+1:k_c}, \mathbf{u}_{j,k_{s,j}+1:k_c}\right\}, \\ \qquad\qquad (\forall i, \forall j) \text{ if } (k_{s,j} < k_c) \\ \left\{\mathsf{bel}(X_{j,k_{s,j}})\right\}, (\forall i, \forall j) \text{ if } (k_{s,j} = k_c) \end{cases} \\ \Rightarrow\ & S_{i,k_e} \supseteq \mathbf{u}_{j,k_c} \text{ or } \mathsf{bel}\left(X_{j,k_c}\right), (\forall i, \forall j) \end{aligned}$$

From $S_{i,k_e} \supseteq S_{j,k_c}$, we expand $S_{j,k_c}(\forall j)$ to show the information that can be found in the knowledge sets depending on whether $k_{s,j}$ (the time of the latest belief for robot j) is less than k_c or equal to k_c. Then we show that either $\mathbf{u}_{j,k_c}$ or $\mathsf{bel}(X_{j,k_c})$ for all robots j will always be available.

Now assuming that the knowledge set of each robot at k_e contains $\mathbf{u}_{j,k_c}$ or $\mathsf{bel}(\mathbf{x}_{j,k_c})$:

$$\begin{aligned} & S_{i,k_e} \supseteq \mathbf{u}_{j,k_c} \text{ or } \mathsf{bel}\left(X_{j,k_c}\right), (\forall i, \forall j) \\ \Rightarrow\ & S_{i,k_e} \supseteq S_{j,k}, (\forall i, \forall j)(k_c \le k \le k_e), \\ \Rightarrow\ & S_{i,k_e} \supseteq S_{j,k_c}, (\forall i, \forall j). \end{aligned}$$

□

Proof of Theorem 2.1 Expressed mathematically, the theorem states that:

$$S_{i,k_{e,i}} \supseteq S_{j,k_{c,i}}, \quad (\forall j)$$
$$\Leftrightarrow S_{i,k_{e,i}} \supseteq (\mathbf{u}_{j,k_{c,i}} \text{ or } \mathsf{bel}(\mathbf{x}_{j,k_{c,i}})), \quad (\forall j)$$

Assume that $C_p(k_{c,i}, k_{e,i})$ exists:

$$S_{i,k_{e,i}} \supseteq S_{j,k_{c,i}}, (\forall j)$$
$$\Rightarrow S_{i,k_{e,i}} \supseteq \begin{cases} \left\{\mathsf{bel}(X_{j,k_{s,j}}), Y_{j,k_{s,j}+1:k_{c,i}} \mathbf{u}_{j,k_{s,j}+1:k_{c,i}}\right\}, \\ \qquad\qquad (\forall j) \text{ if } (k_{s,j} < k_{c,i}) \\ \left\{\mathsf{bel}(X_{j,k_{s,j}})\right\}, (\forall j) \text{ if } (k_{s,j} = k_{c,i}) \end{cases}$$
$$\Rightarrow S_{i,k_{e,i}} \supseteq \mathbf{u}_{j,k_{c,i}} \text{ or } \mathsf{bel}(X_{j,k_{c,i}}), (\forall j)$$

Now assuming that the knowledge set of each robot at $k_{e,i}$ contains $\mathbf{u}_{j,k_{c,i}}$ or $\mathsf{bel}(X_{j,k_{c,i}})$:

$$S_{i,k_{e,i}} \supseteq \mathbf{u}_{j,k_{c,i}} \text{ or } \mathsf{bel}(X_{j,k_{c,i}}), (\forall j)$$
$$\Rightarrow S_{i,k_{e,i}} \supseteq S_{j,k}, (\forall j)(k_{c,i} \le k \le k_{e,i}),$$
$$\Rightarrow S_{i,k_{e,i}} \supseteq S_{j,k_{c,i}}, (\forall j).$$

Therefore, $S_{i,k_{e,i}} \supseteq S_{j,k_{c,i}}, (\forall j) \Leftrightarrow S_{i,k_{e,i}} \supseteq (\mathbf{u}_{j,k_{c,i}}$ or $\mathsf{bel}(X_{j,k_{c,i}})), (\forall j)$ □

Proof of Theorem 2.2 This proof uses the concept of an information flow graph, which models how a network evolves over time, and indicate how information is carried forward in time, and communicated between robots. For more details on information flow graphs, refer to [4]. We approach this proof by examining the knowledge set of each robot and the changes caused by applying the Markov property. We then verify that partial checkpoints continue to exist for all robots.

When a checkpoint exists, and before the Markov property is applied by robot m, the knowledge sets of all robots i contain the belief at some previous time, $k_{s,j}$, for all robots j, as well as odometry and measurements up to k_c:

$$C(k_c, k_e) \text{ exists}$$
$$\Rightarrow C_p(k_c, k_{e,i}) \text{ exists}, (\forall i)$$
$$\Rightarrow S_{i,k_{e,i}} \supseteq S_{j,k_c}, (\forall i, \forall j)$$
$$\Rightarrow S_{i,k_{e,i}} \supseteq \begin{cases} \left\{\mathsf{bel}(\mathbf{x}_{j,k_{s,j}}), \mathbf{u}_{j,k_{s,j}+1:k_c}, Y_{j,k_{s,j}+1:k_c}\right\}, \\ \qquad\qquad (\forall i, \forall j) \text{ if } (k_{s,j} < k_c) \\ \left\{\mathsf{bel}(\mathbf{x}_{j,k_{s,j}})\right\}, (\forall i, \forall j) \text{ if } (k_{s,j} = k_c) \end{cases}$$

It is of interest to know how the knowledge sets of robots that are receiving information from robot m will change once robot m applies the Markov property. Using $\xrightarrow{\text{path}}$ to denote the existence of a path on the information flow graph, let

$$Q = \left\{\text{all robots } i | v(m, k_{e,m}) \xrightarrow{\text{path}} v(i, k_{e,i})\right\}.$$

and $\overline{Q} = N - Q$. Now if we suppose that robot m has applied the Markov property at $k_{e,m}$, then $S_{m,k_{e,m}} \supseteq \mathsf{bel}(\mathbf{x}_j, k_c), (\forall j)$. Furthermore, all robots in Q will also obtain this belief in their knowledge set:

$$S_{i,k_{e,i}} \supseteq \begin{cases} \left\{\mathsf{bel}(\mathbf{x}_{j,k_c})\right\}, (\forall i, \forall j \in Q) \\ \left\{\mathsf{bel}(\mathbf{x}_{j,k_{s,j}}), \mathbf{u}_{j,k_{s,j}+1:k_c}, Y_{j,k_{s,j}+1:k_c}\right\}, \\ \qquad\qquad (\forall i, \forall j \in \overline{Q}) \text{ if } (k_{s,j} < k_c) \\ \left\{\mathsf{bel}(\mathbf{x}_{j,k_{s,j}})\right\}, (\forall i, \forall j \in \overline{Q}) \text{ if } (k_{s,j} = k_c) \end{cases}$$
$$\Rightarrow S_{i,k_{e,i}} \supseteq S_{j,k_c}, (\forall i, \forall j)$$
$$\Rightarrow C_p(k_c, k_{e,i}) \text{ exists}, (\forall i)$$

For robots in $\overline{Q}$, their knowledge sets will remain unaffected and contain the same information as before the Markov property was applied by robot m. Regardless, each of the three cases for $S_{i,k_{e,i}}$ shown above allows us to conclude that by definition, a partial checkpoint exists for all robots i. □

References

1. Kurazume, R., Hirose, S.: An experimental study of a cooperative positioning system. Auton. Robots **8**(1), 43–52 (2000)
2. Roumeliotis, S.I., Bekey, G.A.: Distributed multirobot localization. IEEE Trans. Robot. Autom. **18**(5), 781–795 (2002)
3. Roumeliotis S.I., Rekleitis, I.M.: Propagation of uncertainty in cooperative multirobot localization: analysis and experimental results. Auton. Robots **17**(1), 41–54 (2004)
4. Leung, K.Y.K., Barfoot, T.D., Liu, H.H.T.: Decentralized localization of sparsely-communicating robot networks: a centralized-equivalent approach. IEEE Trans. Robot. Autom. **26**(1), 62–77 (2010)
5. Nerurkar, E., Roumeliotis, S.: Asynchronous multi-centralized cooperative localization. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2010)
6. Leung, K.Y.K., Barfoot, T.D., Liu, H.H.T.: Decentralized simultaneous localization and mapping for dynamic and sparse robot networks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2010)
7. Bar-Shalom, Y.: Update with out-of-sequence measurements in tracking: exact solution. IEEE Trans. Aerosp. Electron. Syst. **38**(3), 769–777 (2002)
8. Bar-Shalom, Y., Chen, H., Mallick, M.: One-step solution for the multistep out-of-sequence-measurement problem in tracking. IEEE Trans. Aerosp. Electron. Syst. **40**(1), 27–37 (2004)
9. Capitán, J., Merino, L., Caballero, F., Ollero, A.: Delayed-state information filter for cooperative decentralized tracking. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3865–3870 (2009)
10. Stachniss, C.: Exploration and Mapping with Mobile Robots. PhD thesis, Universitt Freiburg (2006)
11. Howard, A., Mataric, M.J., Sukhatme, G.S.: Putting the 'i' in 'team': an ego-centric approach to cooperative localization. In: IEEE International Conference on Robotics and Automation (2003)
12. Fenwick, J.W., Newman, P.M., Leonard, J.J.: Cooperative concurrent mapping and localization. In: IEEE International Conferernce on Robotics and Automation (2002)
13. Nettleton, E.W., Durrant-Whyte, H.F., Gibbens, P.W., Goktogan, A.H.: Multiple platform localisation and map building. Proc. SPIE **4196**, 337–347 (2000)
14. Rosencrantz, M., Gordon, G., Thrun, S.: Decentralized sensor fusion with distributed particle filters. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence (2003)
15. Nettleton, E., Thrun, S., Durrant-Whyte, H., Sukkarieh, S.: Decentralised slam with low-bandwidth communication for teams of vehicles. Field & Service Robotic **24**, 179–188 (2006)
16. Reece, S., Roberts, S.: Robust, low-bandwidth, multi-vehicle mapping. In: IEEE FUSION (2005)
17. Thrun, S., Liu, Y., Koller, D., Ng, A., Ghahramani, Z., Durrant-Whyte, H.: Simultaneous localization and mapping with sparse extended information filters. Int. J. Rob. Res. **23**, 693 2004
18. Ko, J., Stewart, B., Fox, D., Konolige, K., Limketkai, B.: A practical, decision-theoretic approach to multi-robot mapping and exploration. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2003)
19. Zhou, X.S., Roumeliotis, S.I.: Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2006)
20. Wang, Z., Huang, S., Dissanayake, G.: Multi-robot simultaneous localization and mapping using d-slam framework. In: IEEE ISSNIP (2007)
21. Howard, A., Sukhatme, G.S., Matarić, M.J.: Multirobot simultaneous localization and mapping using manifold representations. IEEE Special Issue on Multi-Robot Systems **94**(9), 1360–1369 2006
22. Howard, A.: Multi-robot simultaneous localization and mapping using particle filters. Int. J. Rob. Res. **25**(12), 1243–1256 (2006)
23. Simon, D.: Optimal State Estimation. Wiley, New York (2006)
24. Windolf, M., Götzen, N., Morlock, M.: Systematic accuracy and precision analysis of video motion capturing systems—exemplified on the vicon-460 system. J. Biomech. **41**(12), 2776–2780 (2008)
25. Gerkey, B., Vaughan, R.T., Howard, A.: The player/stage project: tools for multi-robot and distributed sensor systems. In: International Conference on Advanced Robotics, pp. 317–323 (2003)
26. Leung, K.Y.K., Halpern, Y., Barfoot, T.D., Liu, H.H.T.: The UTIAS multi-robot cooperative localization and mapping dataset. Int. J. Rob. Res. **30**, 969–974 (2011)
27. Krishnamachari, B., Wicker, S.B., Bejar, R.: Phase transition phenomena in wireless ad hoc networks. In: IEEE Global Telecommunications Conference (2001)
28. Leung, K.Y.K., Barfoot, T.D., Liu, H.H.T.: Distributed and decentralized cooperative simultaneous localization and mapping for dynamic and sparse robot networks. In: IEEE International Conference on Robotics and Automation (2011)